

Generierung von synchronisierten Trajektorien aus geplanten Via- Punkten

Bachelorarbeit

Christian Breu



BACHELORARBEIT

GENERIERUNG VON

SYNCHRONISIERTEN TRAJEKTORIEN

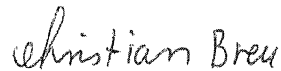
AUS GEPLANTEN VIA-PUNKTEN

Freigabe:

Der Bearbeiter:

Unterschriften

Christian Breu



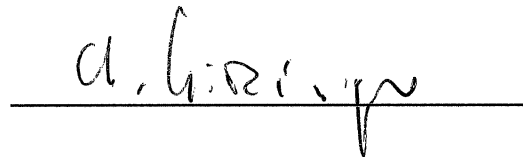
Betreuer:

Andreas Stemmer



Der Institutsdirektor

Prof. Dr. G. Hirzinger



Dieser Bericht enthält 82 Seiten, 55 Abbildungen und 1 Tabelle

**Hochschule für angewandte Wissenschaften –
Fachhochschule Deggendorf**

Fakultät Elektro- und Medientechnik



DLR

**Deutsches Zentrum
für Luft- und Raumfahrt e.V.**
in der Helmholtz-Gemeinschaft



**Hochschule
Deggendorf**

**Generierung von synchronisierten Trajektorien
aus geplanten Via-Punkten**

**Generation of synchronized trajectories
from planned via-points**

Bachelorarbeit zur Erlangung des akademischen Grades:

Bachelor of Engineering
an der Hochschule Deggendorf

vorgelegt von:

Breu, Christian

am: 09.03.2011

Prüfer:

Prof. Dr.-Ing. Nikolaus Müller

Erklärung

Name des Studierenden: Christian Breu
Name des Betreuers: Prof. Dr.-Ing. Nikolaus Müller
Falls extern: Name des Betreuers: Dipl.-Ing. Andreas Stemmer

Thema der Bachelorarbeit:
Generierung von synchronisierten Trajektorien aus geplanten Via-Punkten

1. Ich erkläre hiermit, dass ich die Bachelorarbeit gemäß § 11 APO (Allgemeine Prüfungsordnung der Hochschule für Angewandte Wissenschaften – Fachhochschule Deggendorf) selbständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen oder Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

Deggendorf, _____
(Datum) (Unterschrift des Studierenden)

2. Ich bin damit einverstanden, dass die von mir angefertigte Bachelorarbeit über die Bibliothek der Hochschule einer breiteren Öffentlichkeit zugänglich gemacht wird.

- ☐ Ja
☐ Nein

Falls Ja:

Ich erkläre und stehe dafür ein, dass ich alleiniger Inhaber aller Rechte an der Bachelorarbeit, einschließlich des Verfügungsrechts über Vorlagen an beigelegten Abbildungen, Plänen o. ä. bin und durch deren öffentliche Zugänglichmachung weder Rechte und Ansprüche Dritter noch gesetzliche Bestimmungen verletzt werden.

Deggendorf, _____
(Datum) (Unterschrift des Studierenden)

Nur bei Einverständnis des Verfassers mit einer Veröffentlichung der Bachelorarbeit vom Betreuer auszufüllen:

Eine Aufnahme eines Exemplars der Bachelorarbeit in den Bestand der Bibliothek und die Ausleihe des Exemplars wird

- ☐ befürwortet.
☐ nicht befürwortet.

Deggendorf, _____
(Datum) (Unterschrift des Prüfers)

Danksagung

An dieser Stelle möchte ich mich bei allen Personen bedanken, die mich bei der Anfertigung meiner Bachelorarbeit unterstützt haben.

Mein besonderer Dank gilt Herrn Dipl.-Ing. Andreas Stemmer, der mich als fachlicher Betreuer beim Deutschen Zentrum für Luft- und Raumfahrt in zahlreichen Diskussionen mit Rat und Tat unterstützt und somit maßgeblich zum Gelingen dieser Arbeit beigetragen hat. Schwierige Fragen und Probleme konnten Dank seiner Hilfe beantwortet und gelöst werden.

Des Weiteren bedanke ich mich bei Prof. Dr.-Ing. Nikolaus Müller, meinem Betreuer an der Hochschule Deggendorf, der durch seine Betreuung diese Arbeit ermöglichte. Besonders der unkomplizierte Umgang bei organisatorischen Fragen und sein stets offenes Ohr waren mir eine große Hilfe.

Auch das sehr angenehme Arbeitsklima in der Abteilung mit den stets hilfsbereiten und freundlichen Kollegen erleichterte die Anfertigung der Bachelorarbeit.

Ein besonderer Dank geht an meine Familie, die mir mein Studium ermöglicht und mich vielfältig unterstützt hat. Auch bei meiner Freundin und meinen Freunden möchte ich mich für die motivierenden Worte während der Entstehungszeit dieser Arbeit bedanken.

Inhaltsverzeichnis

1	Einleitung.....	1
1.1	Motivation	1
1.2	Aufgabenstellung	2
1.3	Vorgehensweise	2
2	Grundlagen der Robotik.....	3
2.1	Erklärung von Begriffen	3
2.2	Trajektorienplanung	11
3	Stand der Technik.....	12
3.1	Analytische Ansätze.....	12
3.1.1	Punkt zu Punkt.....	12
3.1.2	Mit Via-Punkt	17
3.1.3	Anwendung auf mehrere Achsen.....	21
3.1.4	Vor- und Nachteile	23
3.1.5	Mögliche Erweiterungen	27
3.1.6	Weitere Funktionen.....	28
3.2	Trajektoriengenerierung mit nichtlinearem Filter.....	31
3.3	Optimale Trajektorienplanung	37
4	Entkoppelte Trajektorienplanung	38
4.1	Eigenschaften	38
4.2	Pfaderzeugung	39
4.3	Berechnung des zeitlichen Verlaufs.....	42
4.3.1	Konstante Skalierung.....	43

4.3.2	Variable Skalierung durch Integration.....	48
4.4	Beispiel einer Trajektorie	53
5	Umsetzung in einen Algorithmus	55
5.1	Polynom-Generator.....	55
5.1.1	Vorgehen zur Pfaderzeugung	55
5.1.2	Bogenstück mit zwei Polynomen dritten Grades.....	61
5.1.3	Erzeugung voll-synchroner Achsen	66
5.1.4	Optimierung der Krümmung im Übergang	68
5.2	Integration	75
5.3	Filter	80
6	Zusammenfassung und Ausblick.....	82
	Literaturverzeichnis	83
	Abbildungsverzeichnis	84

1 Einleitung

1.1 Motivation

Roboter sind aus dem heutigen Alltag nicht mehr wegzudenken. Den größten Anteil besitzen Industrieroboter, die verschiedene Tätigkeiten, wie zum Beispiel Schweißen, Lackieren oder Verpacken verrichten. Aber auch in der Medizin halten Roboter Einzug. Sei es als Träger von Röntgengeräten oder für die Durchführung von minimal-invasiver Chirurgie, durch Roboter sind neuartige, schonendere Eingriffe möglich. Ein weiterer Arbeitsbereich für Roboter ist der Weltraum. Dort können Roboter autonom auf entfernten Planeten Raumstationen errichten oder ferngesteuert von der Erde aus Satelliten reparieren. Zukünftig wird der Bereich von Service-Robotern zunehmend an Bedeutung gewinnen, vor allem im Hinblick auf den bevorstehenden demografischen Wandel und dem damit verbundenen Engpass an Pflegepersonal. Roboter könnten einfache Hol- und Bringdienste verrichten und so beispielsweise bettlägerige Menschen unterstützen.

In allen vorgestellten Bereichen ist es notwendig, dass der Roboter sein Potential voll ausnutzen kann. In der Industrie ist es erwünscht, dass die Tätigkeiten in kürzester Zeit abgewickelt werden, ein Service-Roboter soll sich möglichst menschlich bewegen. Bei beiden Tätigkeiten ist es erforderlich, dass ein Roboterarm seine geplante bzw. vorgegebene Bahn genau verfolgt und nicht von ihr abweicht. Um dies zu erreichen müssen die Gelenke synchron angesteuert werden, die Bewegung eines jeden Gelenkes muss genau die gleiche Zeit benötigen und dabei aber schnellstmöglich vonstatten gehen.

In dieser Bachelorarbeit wird ein Algorithmus entwickelt, der diese Vorgaben erfüllt und als Grundlage für weitere Optimierungen dient.

1.2 Aufgabenstellung

Die Arbeit entstand beim Deutschen Zentrum für Luft- und Raumfahrt in Oberpfaffenhofen am Institut für Robotik und Mechatronik. Ziel dieser Bachelorarbeit ist die Erarbeitung und Implementierung eines Algorithmus, der aus einer Folge von Punkten im Gelenkraum eine ausführbare Trajektorie für den Roboter generiert. Die Trajektorie soll dabei die vorgegebenen Punkte in allen Gelenken synchron mit einer einstellbaren Maximalabweichung erreichen und den Roboter unter Einhaltung von Geschwindigkeits-, Beschleunigungs- und wenn möglich auch Ruckbegrenzungen möglichst schnell zum Zielpunkt bewegen.

1.3 Vorgehensweise

Die Bachelorarbeit umfasst mehrere Arbeitsschritte.

Im ersten Schritt werden die grundlegenden Begriffe der Aufgabenstellung erläutert und kurz auf die Trajektorienplanung eingegangen.

Anschließend wird auf den aktuellen Stand der Technik eingegangen. Dabei gilt es die Vielzahl an derzeit existierenden Ansätzen zu analysieren und zu vergleichen.

Im nächsten Arbeitspaket wird der ausgewählte Ansatz genauer vorgestellt und die dazu notwendige mathematische Vorgehensweise erörtert.

Anschließend müssen nun die erarbeiteten mathematischen Beschreibungen zur Voruntersuchung und zum leichteren Erweitern in einen Algorithmus umgewandelt werden. In diesem Fall wird die plattformunabhängige Software Matlab verwendet.

Abschließend werden die Ergebnisse der Arbeit zusammengefasst und ein Ausblick auf mögliche Erweiterungen gegeben.

2 Grundlagen der Robotik

In diesem Abschnitt werden die grundlegenden Begriffe der Aufgabenstellung erläutert und kurz auf die Trajektorienplanung eingegangen.

2.1 Erklärung von Begriffen

Um ein genaueres Verständnis für die Forderungen der Aufgabenstellung zu erhalten, müssen noch einige Begriffe geklärt werden. Dabei wird unter anderem der Titel dieser Bachelorarbeit „Generierung von synchronisierten Trajektorien aus geplanten Via-Punkten“ aufgearbeitet.

Zunächst wird die Herkunft der „geplanten“ Punkte erläutert. Es kann für einen Roboterarm an jedem rotatorischen Gelenk ein Winkel vorgegeben werden. Bei einem translatorischen Gelenk wird anstatt einem Winkel die Länge des Gelenks vorgegeben. Die Unterscheidung von diesen zwei Bewegungsformen ist für nachfolgende Betrachtungen ohne Bedeutung. Es werden aber aus Gründen der Übersichtlichkeit meistens Gelenkwinkel, also rotatorische Gelenke betrachtet. Diese Winkel sollen in diesem Programm nicht berechnet, sondern lediglich weiter verarbeitet werden. Die Werte, also die Soll-Winkelstellungen, werden bereits von einem übergeordneten Planer geliefert. Dieser Planer generiert für eine geplante Bewegung des Roboters für jedes Gelenk die dazugehörigen Winkelstellungen. Der Planer kann dabei ein eigenes Programm sein, welches zum Beispiel selbstständig durch Bildverarbeitung einen Weg für den Roboterarm durch den Raum ermittelt. Möglich ist aber auch, dass ein Benutzer Werte vorgibt, die der Roboterarm einnehmen soll.

Zur Veranschaulichung des Zusammenhangs eines geplanten Weges mit den dazugehörigen Gelenkwinkeln wird als Beispiel ein Roboter mit zwei rotatorischen Achsen herangezogen. Dabei ist ein Motor für das erste Gelenk, ein zweiter für das zweite Gelenk verantwortlich. Die beiden Gelenkwinkel werden mit q_1 und q_2 bezeichnet. Bei Robotern mit n Achsen heißt der Gelenkwinkel in der letzten Achse q_n . In dem unten dargestellten Beispiel soll sich der Endpunkt

des Roboters, also der Endeffektor, vom Startpunkt zum Endpunkt bewegen. Dazu muss nur das zweite Gelenk verändert werden, da beide Punkte auf einem Kreis liegen, dessen Mittelpunkt das zweite Gelenk ist. Der Gelenkwinkel q_1 bleibt während des gesamten Bewegungsvorgangs auf einem konstanten Wert. Nur der Winkel q_2 muss sich vom Istwert zu einer neuen Winkelstellung bewegen, um den Endpunkt zu erreichen. Der tatsächlich nötige Wert für q_2 , damit der Roboter diese Position einnimmt, ist von allen vorhandenen Achsen und deshalb vom Aufbau des Roboters abhängig. Die Umrechnung auf die Position des Endeffektors anhand des Roboteraufbaus ist durch die Rücktransformation beschrieben.

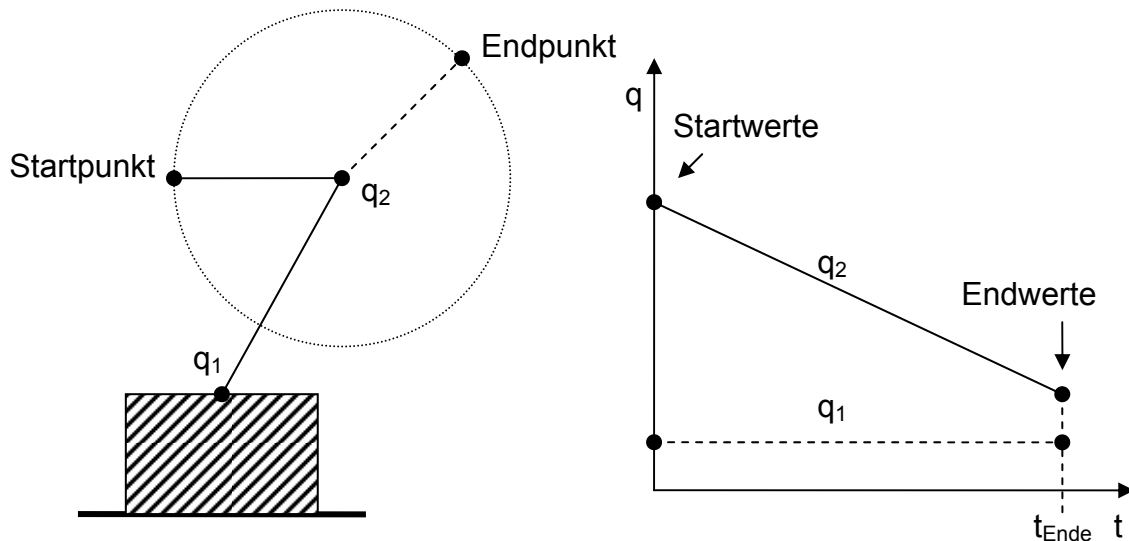


Abbildung 2.1: Zusammenhang Gelenkwinkel zur Bewegung im Raum

Der Planer liefert allerdings nicht den gesamten Verlauf für die Gelenkwinkel, sondern nur die neuen benötigten Sollwerte für das Einnehmen der Endposition. Es liegen also lediglich zwei neue Werte für q_1 und q_2 vor. In dem in Abbildung 2.1 auf der rechten Seite dargestellten Koordinatensystem ist über die x-Achse die Zeit t , über die y-Achse die Gelenkwinkel q aller Achsen aufgetragen. Da in dieser Arbeit die Gelenkwinkel von großer Bedeutung sind und diese für Berechnungen und zur Veranschaulichung in einen Graphen eingetragen werden, dient der Aufbau dieses Koordinatensystems als Grundlage für alle weiteren Betrachtungen.

Für komplexe Bewegungsabläufe liefert der Planer aber nicht nur zwei, sondern gleich mehrere Werte. Dies ist schon bei relativ einfachen Bewegungsabläufen nötig. Soll der Roboter beispielsweise um eine Ecke fahren, so sind mindestens drei Winkelstellungen nötig. Dies sind der Anfangs- und Endwert und der Winkel im Eckpunkt. Diese Soll-Winkelstellung zwischen dem Start- und dem Zielwert wird Via-Punkt genannt [6]. In der Literatur existieren unterschiedliche Definitionen für einen Via-Punkt. Zum einen gibt es die Vorgabe, dass die Trajektorie genau durch diesen Punkt verlaufen soll. Dabei soll die Geschwindigkeit, mit der sich der Antrieb dreht, aber nicht null werden, wie es zum Beispiel der Fall wäre, wenn zwei Punkt zu Punkt Verbindungen nacheinander ausgeführt werden [6]. Eine andere Definition gibt an, dass diese Winkelstellung im Via-Punkt nicht genau eingenommen, sondern nur angenähert wird, die Geschwindigkeit ist hierbei aber auch nicht null [6]. In dieser Arbeit dient die zweite Definition als Grundlage, jedoch mit dem Zusatz, dass eine maximale Abweichung vorgegeben werden kann.

Zur Veranschaulichung eines Via-Punktes wird als Beispiel ein x-y-Portal, ähnlich dem Aufbau eines Krans in einer Halle, herangezogen. Dabei ist ein Motor für die Bewegung in x-Richtung, ein zweiter für die Bewegung in y-Richtung verantwortlich. Das Portal soll sich von einem Startpunkt über einen Via-Punkt zum Endpunkt bewegen. Die Motoren an den Antriebsachsen müssen dabei den Winkel vom Istwert zum ersten Sollwert für den Via-Punkt und anschließend zum zweiten Sollwert für den Endpunkt einnehmen. Schematisch dargestellt ist ein solches Portal in Abbildung 2.2. Eingezeichnet sind dabei der Verfahrensweg mit Via-Punkt und die dazu nötigen Gelenkwinkel. Die Winkelstellung q_1 gehört zu dem ersten Motor, q_2 ist für den zweiten. Die Istwerte der beiden Antriebe sind bei einer Zeit von $t = 0$ eingezeichnet. Die Soll-Winkelstellungen für den Via-Punkt sind über den Zeitpunkt t_{via} , für den Endwert über t_{Ende} eingetragen.

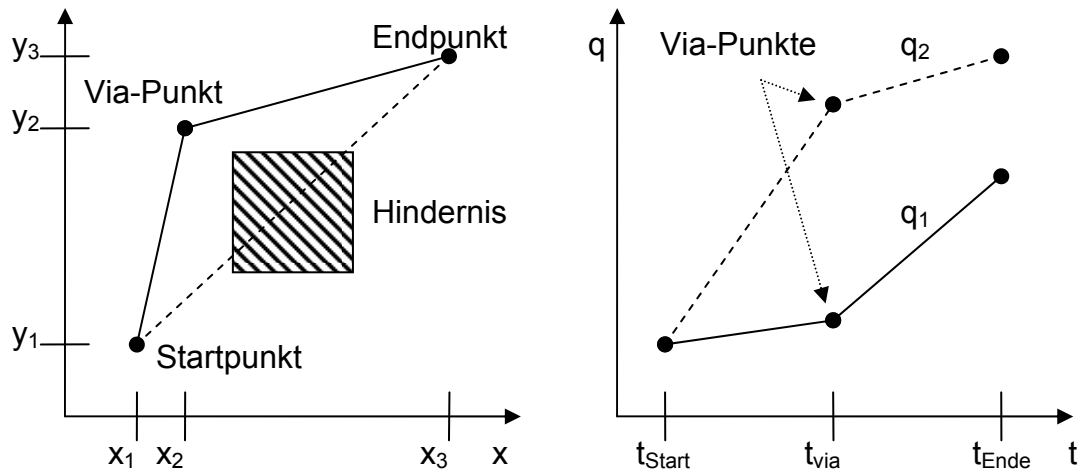


Abbildung 2.2: Bahnverlauf eines x-y-Portals mit Via-Punkt

Da in dieser Arbeit der Via-Punkt so definiert ist, dass eine maximale Abweichung vorgegeben werden kann, muss noch kurz auf die Auswirkung einer Toleranz eingegangen werden. Auch für dieses Beispiel wird ein x-y-Portal zu Rate gezogen. Die Toleranz wird für die Soll-Gelenkwinkel vorgegeben. Bei dem Portal sind zwei Gelenkwinkel für die Position verantwortlich. Werden nun für beide Achsen Toleranzen zugelassen, so ergibt sich aus einem einzigen Startpunkt ein ganzer Bereich. Bei gleicher Dimensionierung der Achsen und gleichen Toleranzen ist dieser Bereich quadratisch. In der folgenden Abbildung sind dies die grauen Quadrate.

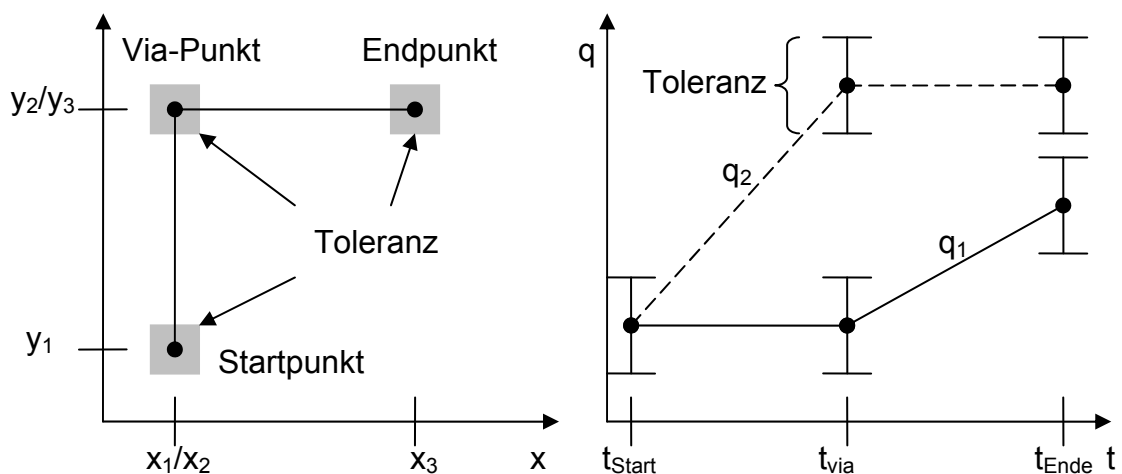


Abbildung 2.3: Positionsabweichungen durch Toleranzen im Gelenkwinkel

Die Toleranz soll nicht nur an den einzelnen Punkten, die der Planer vorgibt, eingehalten werden soll, sondern auch in den Übergangsbereichen. Deshalb

wird im Koordinatensystem, in dem die Winkel eingetragen werden, nicht nur die Abweichung an den Punkten, sondern auch dazwischen begrenzt. Dadurch ergibt sich ein Toleranzband, welches nicht verlassen werden darf. Die Darstellung des Toleranzbandes im Koordinatensystem und des dazugehörigen möglichen Bereichs, den das Portal einnehmen kann, ist nachfolgend dargestellt. Beispielhaft ist die Auswirkung auf den Verlauf des Pfades, wenn die Gelenkwinkel nicht exakt eingehalten werden, in rot eingezeichnet. Dieses Toleranzband ist von großer Bedeutung, da es sehr viele mathematische Möglichkeiten gibt, um zwei Punkte, also zwei Gelenkwinkel im Koordinatensystem, miteinander zu verbinden. Mögliche Funktionen sind in Abbildung 2.5 dargestellt. Die gestrichelten Verbindungen halten dieses Toleranzband eventuell nicht ein.

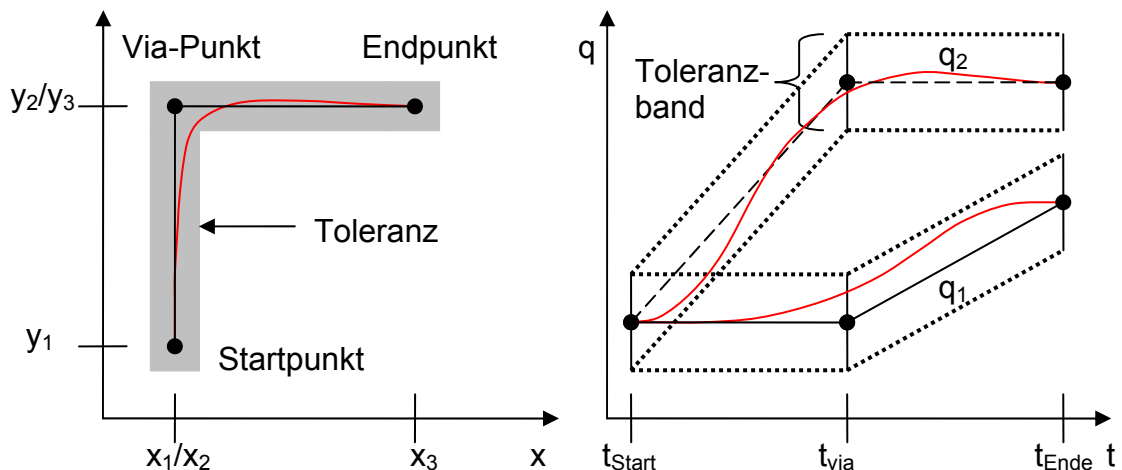


Abbildung 2.4: Zulässiges Toleranzband zum Einhalten einer vorgegebenen Abweichung

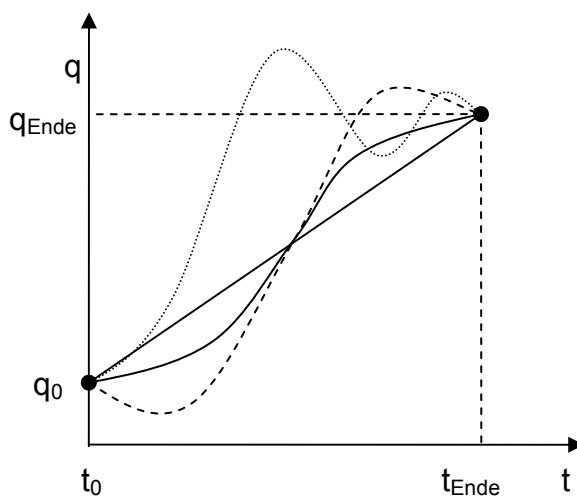


Abbildung 2.5: Verschiedene Möglichkeiten für Punkt zu Punkt Verbindungen

Als nächstes folgt die Erklärung, weshalb überhaupt Trajektorien generiert werden müssen. Wie soeben beschrieben, dienen als Ausgangslage verschiedene Winkelwerte für jede Achse des Roboters. Im einfachsten Fall soll sich der Roboter von einem Startpunkt zu einem Endpunkt bewegen. Der Planer liefert also zunächst einen Startwinkel in einem Robotergetriebe. Im Normalfall ist der Startwert auch der Istwert des Gelenks. Der zweite Wert, den der Planer liefert, ist der Gelenkwinkel, den der Roboter einnehmen muss, um zum Endpunkt zu gelangen, also der Sollwert. Es sind zwei Winkelstellungen für ein Gelenk vorgegeben, die sicherstellen, dass der Roboter im Raum genau der geplanten Bahn folgt. Der Motor, der die Achse antreibt, muss so angesteuert werden, dass sich diese zur Endstellung dreht. Derzeitige Motorregler benötigen jede Millisekunde einen Sollwert. Wird diesem Regler der neue Soll-Winkel übergeben, so bedeutet dies eine sprunghafte Änderung des Sollwerts. Da der neue Wert möglichst schnell erreicht werden soll, ist es üblich, dass der Regler einen Überschwinger erzeugt, also die Ausgangsgröße den Sollwert übersteigt. Ein Verhindern des Überschwingens bedeutet, dass der neue Sollwert asymptotisch erreicht wird, was allerdings erheblich mehr Zeit benötigt. Besser ist es nicht einfach nur die neue Winkelstellung zu übergeben, sondern aus dem Ist- und dem Sollwert eine stetige Funktion zu generieren, die zu jedem Abtastzeitpunkt des Reglers eine neue Soll-Winkelstellung liefert, so dass die Fähigkeiten des Antriebs optimal genutzt werden. Diese Funktion beschreibt nicht nur den zeitlichen Verlauf der Winkelstellungen, sondern durch die erste und zweite Ableitung auch die Geschwindigkeit, mit der sich die Motoren drehen sollen, sowie die Beschleunigung, mit denen diese beaufschlagt werden. Die dritte Ableitung entspricht dem Ruck j .

Für die Forderung von „synchronisierten Trajektorien“ gilt es die verschiedenen Arten der Synchronisation zu erörtern. Diese Betrachtung ist erst bei der Planung von mehreren Achsen notwendig. Bei der Bewegung eines Roboterarms werden nicht nur die Winkelstellungen in einer Achse, sondern meistens sogar in jeder Achse verändert. In diesem Fall wird für jeden Motorregler an jeder Achse eine Trajektorie geplant. Soll eine jede Achse genau zum gleichen Zeitpunkt mit der Bewegung beginnen und auch wieder enden, so spricht man von

halb-synchron bzw. von zeit-synchronen Trajektorien [7]. Voll-synchron bedeutet, dass eine jede Trajektorie nicht nur die gleiche Gesamtzeit benötigt, sondern dass alle Bewegungsphasen gleich lange dauern. Das bedeutet, dass alle Achsen genau die gleiche Zeit zum Beschleunigen und zum Verzögern benötigen, sie bewegen sich auch exakt die gleich lange Zeit mit konstanter Geschwindigkeit. Zur Verdeutlichung der unterschiedlichen Auswirkungen von asynchronen, zeit-synchronen und voll-synchronen Trajektorien wird als Beispiel wieder ein x-y-Portal herangezogen. Die Motoren an den Antriebsachsen müssen dabei den Winkel von einem Startwert zu einem Endwert ändern, um das Portal vom Startpunkt zum Endpunkt zu bewegen. Eingezeichnet sind in Abbildung 2.6 die verschiedenen Fahrwege, die sich mit unterschiedlicher Synchronität ergeben. Der Verlauf mit der Nummer eins kommt zustande, wenn zuerst Motor eins und erst nach Beenden dieses Fahrvorgangs Motor zwei mit dem neuen Sollwinkel angesteuert wird. Typisch für die Ansteuerung mit asynchronen Trajektorien ist dabei der Verlauf zwei. Die Bewegung in y-Richtung wird zuerst beendet, die Bewegung in x-Richtung ist dabei noch nicht abgeschlossen. Zeit-synchron ist der Verlauf der Bahn drei. Hier beginnen und beenden beide Antriebe zum gleichen Zeitpunkt ihre Bewegung, jedoch sind die einzelnen Bewegungsphasen nicht synchron. Die Bewegung in x-Richtung verläuft schon konstant, während sich der Antrieb in y-Richtung noch in der Beschleunigungsphase befindet. Erst im Bewegungsverlauf Nummer vier, welcher eine direkte Gerade zwischen dem Start- und Endpunkt beschreibt, handelt es sich um eine Ansteuerung mit voll-synchronen Trajektorien. Beide Antriebe benötigen die gleiche Zeit für den Beschleunigungsvorgang. Die Phase, in der die Geschwindigkeit einen konstanten Wert besitzt, dauert ebenso bei beiden Achsen gleich lange. Auch der Verzögerungsvorgang ist zeitlich synchron. Die Abbildung verdeutlicht, dass sich unterschiedliche Synchronitäten gravierend auf den realen Bahnverlauf eines Roboters auswirken. Der Wegplaner geht aber immer von einer direkten Verbindung von Start- und Endpunkt aus. Werden also nicht voll-synchrone Trajektorien geplant, so ist es möglich, dass der Roboter von der Bahn abweicht und dadurch eventuell Hindernisse

durchfahren würde, was zu einer Beschädigung des Roboters oder des Hindernisses führen kann.

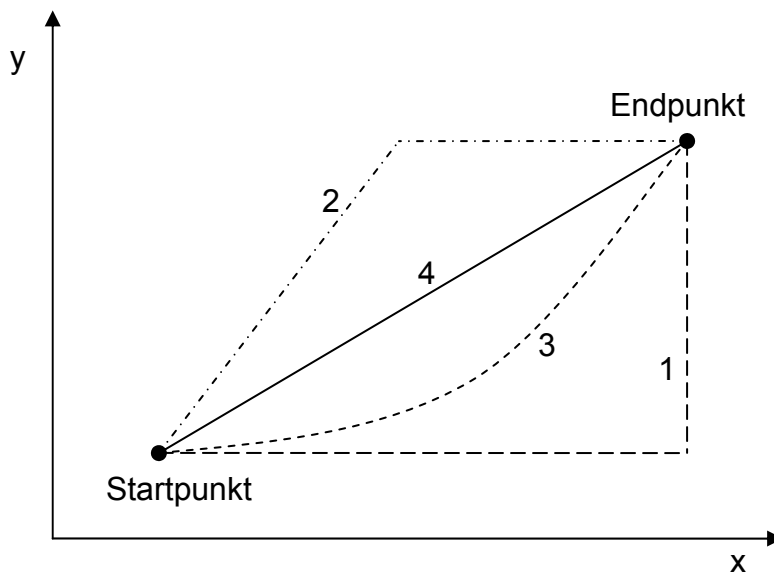


Abbildung 2.6: Bahnverläufe mit unterschiedlichen Synchronitäten

Die Schwierigkeiten, die bei der Generierung von synchronisierten Trajektorien auftreten, sind im Abschnitt 3.1.3 beschrieben. Es sei aber bereits jetzt erwähnt, dass voll-synchrone Trajektorien zeitlich gesehen nicht der schnellsten Verbindung von Start- und Endpunkt entsprechen. Halb-synchrone Trajektorien wie Nummer zwei und drei sind meist schneller.

Neben den Bedingungen, die sich aus dem Titel der Bachelorarbeit ergeben, gibt es aber noch weitere Eigenschaften, die erfüllt werden müssen. Diese wären zum Beispiel, dass für die Trajektorienplanung die Grenzen der Antriebe berücksichtigt werden müssen. Dies sind im Besonderen die maximale Gelenkwinkelgeschwindigkeit, also wie schnell sich ein Antrieb drehen kann, um den Gelenkwinkel zu verändern. Außerdem muss die maximale Beschleunigung des Motors eingehalten werden. Wünschenswert wäre auch noch eine Begrenzung des Rucks, also der Ableitung der Beschleunigung, dies kann aber als optional betrachtet werden. Daraus folgt, dass die Trajektorie glatt, also stetig differenzierbar sein muss. Diese Forderung berücksichtigt, dass der Motor keinen Sprung in der Geschwindigkeit oder dem Gelenkwinkel ausführen kann, die Änderung der Geschwindigkeit ist immer stetig. Für eine Begrenzung des

Rucks muss die Trajektorie dreimal ableitbar sein. Die zweite Ableitung, also die Beschleunigung muss dann stetig, aber nicht differenzierbar sein.

2.2 Trajektorienplanung

Prinzipiell kann zwischen Pfadplanung und Trajektorienplanung unterschieden werden. Vor allem in Kapitel 4 bei der entkoppelten Trajektorienplanung ist diese Unterscheidung sehr wichtig, weshalb schon jetzt kurz darauf eingegangen wird. Die Pfadplanung legt die geometrische Bahn fest, auf der sich der Roboter zwischen einem Start- und einem Zielpunkt bewegen soll. Die Trajektorienplanung beschreibt den Verlauf der Geschwindigkeit und Beschleunigung, mit der der Roboter entlang der Bahn verfahren soll. Das heißt bei der Pfadplanung wird der geometrische Verlauf der Bahn unabhängig von der Geschwindigkeit bzw. des zeitlichen Verlaufs festgelegt [8]. In den meisten nachfolgend vorgestellten Ansätzen wird aber sofort versucht das Erreichen des vorgegebenen Winkels zeitabhängig zu bestimmen, es wird also direkt eine Trajektorie geplant. Im Koordinatensystem, in dem die Gelenkwinkel eingezeichnet sind, ist die Unterscheidung zwischen Pfad- und Trajektorienplanung an der Beschriftung der x-Achse zu erkennen. Wird eine Trajektorie geplant, so repräsentiert die x-Achse die Zeit t , bei der Pfadplanung werden die Gelenkwinkel über den Bahnparameter s aufgetragen.

3 Stand der Technik

Für die Trajektorien- bzw. die Pfadplanung gibt es bereits eine Vielzahl von Ansätzen. In diesem Kapitel wird ein Überblick über die meisten derzeit in der Literatur vertretenen Lösungsvorschläge gegeben. Sie werden in kurzen Beispielen erläutert und es wird außerdem auf die Vor- und Nachteile eingegangen. Anhand dieser Untersuchung soll verdeutlicht werden, weshalb die Wahl auf den Ansatz der entkoppelten Trajektorienplanung gefallen ist.

3.1 Analytische Ansätze

Zur Veranschaulichung aller hier vorgestellten Verfahren werden die Gelenkwinkel wieder in Koordinatensystemen dargestellt. Wie in Abbildung 2.6 im Kapitel 2.1 dargestellt, soll eine mathematische Funktion gefunden werden, die zwei Punkte, welche verschiedene Winkelstellungen eines Roboter gelenkes repräsentieren, miteinander verbinden. Das heißt das Gelenk dreht sich von einer vorgegebenen Ausgangslage in eine gewünschte Endlage. Bei einigen Beispielen sollen zwischen diesem Verfahrensweg weitere Winkelstellungen eingenommen werden. Dies ist vor allem dann nötig, wenn z. B. der Roboterarm eine weite Strecke zurücklegt und dabei ein Hindernis umfahren soll. Diese gewünschten Winkel zwischen der Anfangs- und der Endposition werden Via-Punkte genannt. Das Gelenk soll also verschiedene vorgegebene Positionen einnehmen. Diese Bedingungen werden im Kapitel 2.1 abgehandelt. Gültig für alle Ansätze ist die Bedingung, dass die Geschwindigkeit des Gelenks am Anfang und am Ende null ist, es sich also aus dem Stillstand heraus in Rotation versetzt, und am Ende wieder zum Stehen kommt. Der Gelenkwinkel wird mit q bezeichnet, die Gelenkwinkelgeschwindigkeit \dot{v} oder \dot{q} , die Gelenkwinkelbeschleunigung a bzw. \ddot{q} .

3.1.1 Punkt zu Punkt

Die einfachste Aufgabenstellung bei der Trajektorienplanung ist die Verbindung von Punkt zu Punkt. Die einfachste mathematische Funktion, die zwei Punkte

miteinander verbindet, ist eine Gerade. Da diese Gerade überall die gleiche Steigung und damit eine konstante Geschwindigkeit aufweist, wird die Bedingung, dass Start- und Endgeschwindigkeit null sein soll, nicht erfüllt. Es wird eine Funktion benötigt, welche vier Bedingungen erfüllt. Nämlich das Erreichen der vorgegebenen Start- und Endwinkel sowie die, dass Start- und Endgeschwindigkeit, also die 1. Ableitung der Funktion, null ist. Um diese vier Bedingungen zu erfüllen ist eine Funktion mit vier freien Koeffizienten nötig, also z. B. ein Polynom dritten Grades. Siehe hierzu Formel (3.1). Wird dazu noch die Zeit vorgegeben die benötigt wird um vom Startpunkt zum Endpunkt zu gelangen, so kann das Polynom folgendermaßen berechnet werden [6].

Ein kubisches Polynom hat die Form:

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \quad (3.1)$$

Die erste und zweite Ableitung ergeben sich zu:

$$\begin{aligned} \dot{q}(t) &= a_1 + 2a_2 t + 3a_3 t^2 \\ \ddot{q}(t) &= 2a_2 + 6a_3 t \end{aligned} \quad (3.2)$$

Das Erreichen der Start- und Endwinkel q_a und q_e ergibt folgende Ausdrücke:

$$\begin{aligned} q(0) &= q_a \\ q(t_{ende}) &= q_e \end{aligned} \quad (3.3)$$

Die Vorgabe der Start- und Endgeschwindigkeit von 0 °/s:

$$\begin{aligned} \dot{q}(0) &= 0 \\ \dot{q}(t_{ende}) &= 0 \end{aligned} \quad (3.4)$$

Dadurch ergeben sich folgende Gleichungen:

$$\begin{aligned} q_a &= a_0 \\ q_e &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 \\ 0 &= a_1 \\ 0 &= a_1 + 2a_2 t_{ende} + 3a_3 t_{ende}^2 \end{aligned} \quad (3.5)$$

Aufgelöst ergibt sich dann für die 4 Koeffizienten:

$$\begin{aligned} a_0 &= q_a \\ a_1 &= 0 \\ a_2 &= \frac{3}{t_{ende}^2} (q_e - q_a) \\ a_3 &= -\frac{2}{t_{ende}^3} (q_e - q_a) \end{aligned} \quad (3.6)$$

Geplottet für eine Zeit von 4 Sekunden ergibt sich dabei der nachfolgend dargestellte Verlauf.

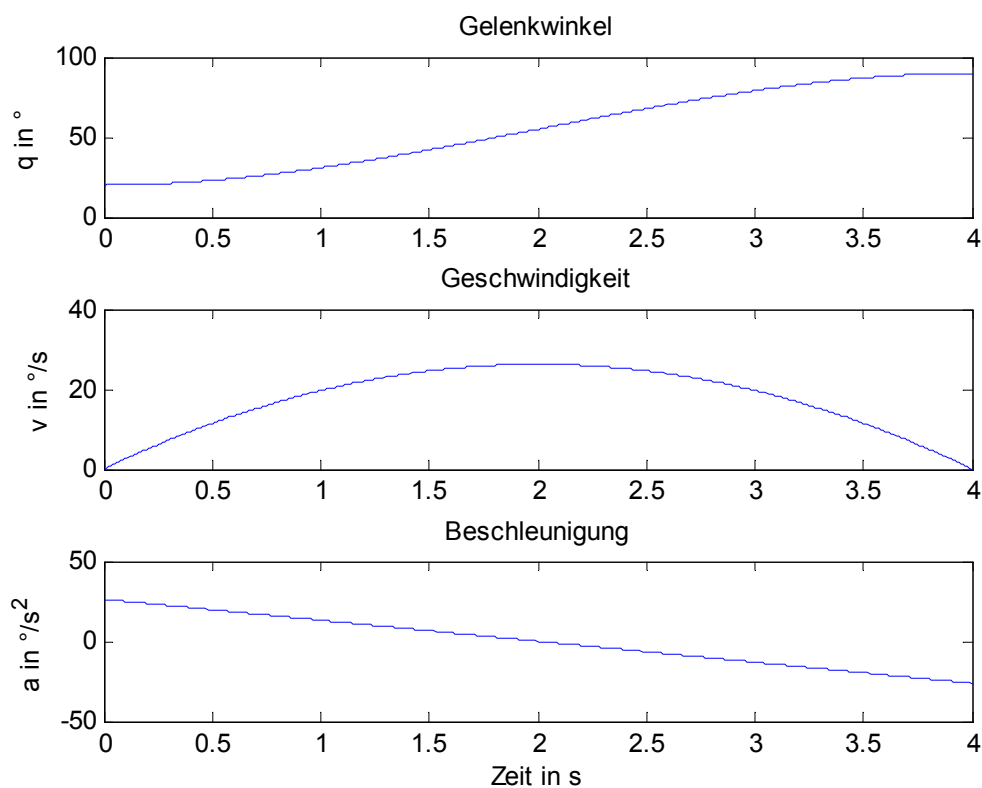


Abbildung 3.1: Punkt zu Punkt Verbindung mit Polynome dritten Grades

Der Verlauf des Gelenkwinkels beginnt und endet bei den gewünschten Werten, die Geschwindigkeit, also die 1. Ableitung ist in diesen Punkten Null. Für die Zeit wurde $t = 4$ s vorgegeben. Außerdem ist noch die 2. Ableitung abgebildet, also der Verlauf der Beschleunigung. Allgemein kann die Aussage getroffen werden, dass der Verlauf des Gelenkwinkels sehr glatt ist. Als größter Nachteil bei diesem Ansatz gilt, dass die Zeit vorgegeben werden muss. Der

Spitzenwert der Gelenkwinkelgeschwindigkeit, als auch die Spitzenwerte bei der Beschleunigung, ergeben sich aufgrund der Vorgabe der Zeit. Es könnte zwar, durch Auflösen von (3.2), eine Zeit ermittelt werden, die entweder eine maximale Geschwindigkeit oder die Beschleunigung berücksichtigt, beides ist jedoch nicht möglich. Genau dies ist unerwünscht, es sollen die maximal zulässigen Werte für die Beschleunigung und die Geschwindigkeit des Gelenkes vorgegeben werden können. Die Zeit soll, mit der Bedingung möglichst kurz zu dauern, aus diesen Größen ermittelt werden.

Ein anderer gebräuchlicher Ansatz für die Punkt zu Punkt Verbindung ist der Ansatz einer Geraden mit parabolischen Enden [6]. Siehe hierzu Abbildung 3.2. Dabei setzt sich die Funktion aus parabolischen Teilstücken, welche für die Beschleunigung zuständig sind, und aus einer Geraden, welche für eine konstante Geschwindigkeit sorgt, zusammen. Die parabolischen Teile sind in blau, das gerade Teilstück ist in rot eingezeichnet. Auch bei dieser Funktion muss für die Berechnung die Zeit vorgegeben werden, die Geschwindigkeit wird daraus ermittelt. Ein Vorteil bei diesem Ansatz ist allerdings, dass die Beschleunigung schon begrenzt werden kann. Durch Variation der maximalen Beschleunigung a_{\max} kann die Dauer der Beschleunigung und der Verzögerung beeinflusst werden. Ein größeres a_{\max} verkürzt diese Vorgänge, eine Verkleinerung bewirkt eine Verlängerung. Die Berechnung der einzelnen Funktionen ist nachfolgend aufgeführt.

Die Geschwindigkeit am Ende des Parabelbogens muss gleich groß der Geschwindigkeit des linearen Teils sein.

$$\ddot{q}t_b = \frac{q_h - q_b}{t_h - t_b} \quad (3.7)$$

Der Wert von q_b ist gegeben durch:

$$q_b = q_0 + \frac{1}{2} \ddot{q}t_b^2 \quad (3.8)$$

Der Hilfspunkt q_h kann ermittelt werden mit:

$$q_h = \frac{q_f + q_0}{2} \quad (3.9)$$

Durch Kombination von (3.7) mit (3.8), einsetzen von (3.9) sowie der Bedingung $t = 2t_h$ ergibt sich:

$$\ddot{q}t_b^2 - \ddot{q}t_b + (q_f - q_0) = 0 \quad (3.10)$$

Dabei ist t die gesamte Zeit für die Bewegung, im folgenden Beispiel ist $t = 4$ s. Die 2. Ableitung von q , also die Beschleunigung kann vorgegeben werden. Damit eine Lösung existiert, muss diese groß genug sein. Mit diesen Werten kann die Dauer des Beschleunigungsvorgangs berechnet werden.

$$t_b = \frac{t}{2} - \frac{\sqrt{\ddot{q}^2 t^2 - 4\ddot{q}(q_f - q_0)}}{2\ddot{q}} \quad (3.11)$$

Die Beschleunigung muss dabei folgende Bedingung erfüllen. Beträgt die Beschleunigung genau gleich der Bedingung (3.12), schrumpft der lineare Anteil auf die Länge Null. Die Kurve besteht nur aus einem Beschleunigungs- und Verzögerungsvorgang.

$$\ddot{q} \geq \frac{4(q_f - q_0)}{t^2} \quad (3.12)$$

Geplottet ergibt sich folgender Verlauf.

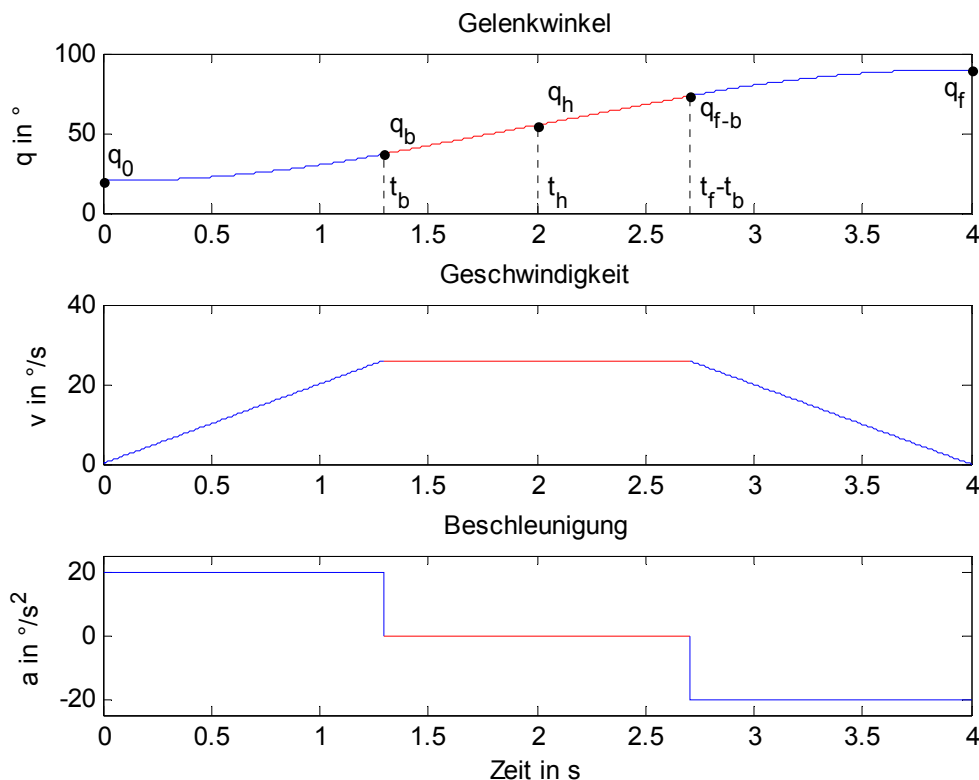


Abbildung 3.2: Punkt zu Punkt Verbindung durch Gerade mit parabolischen Enden

3.1.2 Mit Via-Punkt

Bei den beiden in Abschnitt 3.1.1 vorgestellten Ansätzen wird der Gelenkwinkel nur von einem Anfangswert auf einen Endwert verstellt. Im Folgenden werden diese beiden Ansätze so erweitert, dass auch vorgegebene Werte zwischen diesen beiden Zuständen, so genannte Via-Punkte, eingenommen werden.

Als erstes wird das Polynom dritten Grades betrachtet. Um nun also einen weiteren Punkt im Gelenkwinkelraum einzunehmen, müsste entweder die Ordnung des Polynoms erhöht werden oder es werden mehrere Polynome dritter Ordnung aneinander gefügt [6]. Im Folgenden wird davon ausgegangen, dass ein Via-Punkt vorhanden ist. Dafür werden zwei Polynome dritten Grades verwendet, welche folgende Bedingungen erfüllen müssen. Die Geschwindigkeit muss am Anfangs- und am Endpunkt null sein, im Via-Punkt, also dort wo sich die zwei Polynome treffen, muss die Geschwindigkeit stetig differenzierbar sein.

Die Beschleunigung muss im Übergangspunkt stetig sein. Des Weiteren soll das erste Polynom den Startwert und den Via-Punkt erreichen, das zweite soll vom Via-Punkt bis zum Endpunkt gehen. Somit ergeben sich zwei Funktionen dritten Grades mit je vier Koeffizienten. Da oben genau acht Bedingungen beschrieben wurden, können die insgesamt acht Koeffizienten berechnet werden. Dies ist nachfolgend dargestellt [6].

Die erste Funktion berechnet sich mit:

$$q(t) = a_{10} + a_{11}t + a_{12}t^2 + a_{13}t^3 \quad (3.13)$$

Die zweite ist:

$$q(t) = a_{20} + a_{21}t + a_{22}t^2 + a_{23}t^3 \quad (3.14)$$

Jede Formel beginnt bei $t = 0$ und endet bei $t = t_{\text{ende}}$

$$\begin{aligned} q_0 &= a_{10} \\ q_v &= a_{10} + a_{11}t_{\text{ende1}} + a_{12}t_{\text{ende1}}^2 + a_{13}t_{\text{ende1}}^3 \\ q_v &= a_{20} \\ q_e &= a_{20} + a_{21}t_{\text{ende2}} + a_{22}t_{\text{ende2}}^2 + a_{23}t_{\text{ende2}}^3 \\ 0 &= a_{11} \\ 0 &= a_{21} + 2a_{22}t_{\text{ende2}} + 3a_{23}t_{\text{ende2}}^2 \\ a_{11} + 2a_{12}t_{\text{ende1}} + 3a_{13}t_{\text{ende1}}^2 &= a_{21} \\ 2a_{12} + 6a_{13}t_{\text{ende1}} &= 2a_{22} \end{aligned} \quad (3.15)$$

Aufgelöst mit der Bedingung t_{ende1} ist genau so lang wie t_{ende2} ergibt sich so für die Koeffizienten:

$$\begin{aligned}
 a_{10} &= q_0 \\
 a_{11} &= 0 \\
 a_{12} &= \frac{12q_v - 3q_e - 9q_0}{4t_{\text{ende}}^2} \\
 a_{13} &= \frac{-8q_v + 3q_e + 5q_0}{4t_{\text{ende}}^3} \\
 a_{20} &= q_v \\
 a_{21} &= \frac{3q_e - 3q_0}{4t_{\text{ende}}} \\
 a_{22} &= \frac{-12q_v + 6q_e + 6q_0}{4t_{\text{ende}}^2} \\
 a_{23} &= \frac{8q_v - 5q_e - 3q_0}{4t_{\text{ende}}^3}
 \end{aligned} \tag{3.16}$$

Wird nun wieder eine Zeit für jede Funktion von vier Sekunden vorgegeben, so ergibt sich der in Abbildung 3.3 dargestellte Verlauf. Dabei ist in blau die erste, in rot die zweite Funktion dargestellt. Im Übergang der beiden Funktionen ist der Via-Punkt q_v . Bei diesem Vorgehen ergeben sich allerdings die gleichen Nachteile wie bei dem Polynom dritter Ordnung ohne Via-Punkt. Die Zeit muss vorgegeben werden, sowohl der Spitzenwert der Gelenkwinkelgeschwindigkeit als auch die Spitzenwerte bei der Beschleunigung ergeben sich aufgrund der vorgegebenen Zeit. Die Maximalwerte können nicht begrenzt werden.

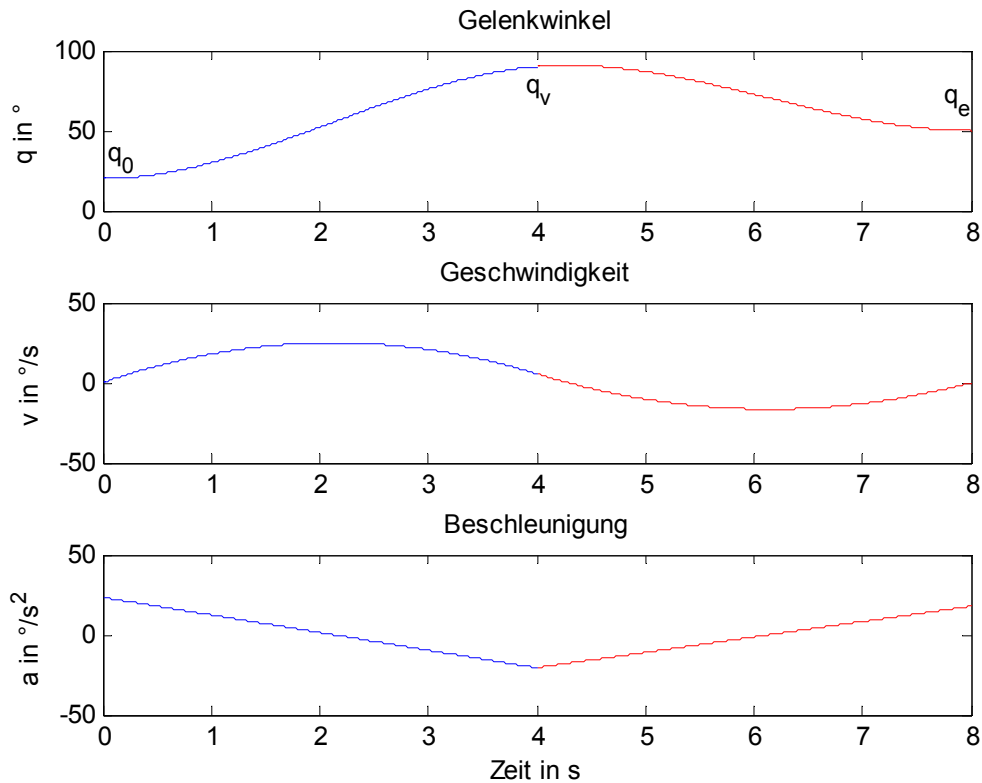


Abbildung 3.3: Verbindung mit Via-Punkt mit Polynome dritten Grades

Bei dem Verfahren mit Geraden und parabolischen Enden ist folgendes Vorgehen üblich [6]. Der erste und der letzte Parabelbogen werden äquivalent der Berechnung der Parabelbögen ohne Via-Punkt durchgeführt. Für die Via-Punkte wird eine andere Vorgehensweise angewandt. Im Beispiel in Abbildung 3.4 werden die Punkte zwei und drei sowie die Punkte drei und vier mit einer Geraden verbunden. Die Steigungen dieser Geraden werden entweder über die Vorgabe der Zeit oder durch Vorgabe einer maximalen Winkelgeschwindigkeit berechnet, wobei die Steigung die Geschwindigkeit repräsentiert.

Die Steigung m berechnet sich mit Vorgabe der Zeit folgendermaßen:

$$m = \frac{\Delta q}{\Delta t} \quad (3.17)$$

Vorgabe der maximalen Gelenkwinkelgeschwindigkeit v ergibt die Zeit t .

$$t = \frac{\Delta q}{v_{max}} \quad (3.18)$$

Das heißt bei diesem Ansatz kann bereits eine Maximalgeschwindigkeit vorgegeben werden, die Zeit wird daraus ermittelt. Üblich ist die Vorgabe der Zeit. Die Vorgabe der maximalen Beschleunigung hat Einfluss auf die Krümmung der Parabelbögen und ist somit für die Verbindungsstücke zwischen den Geraden zuständig. Im Übergang von der Geraden auf die Parabel muss als Bedingung berücksichtigt werden, dass die Geschwindigkeit gleich groß ist.

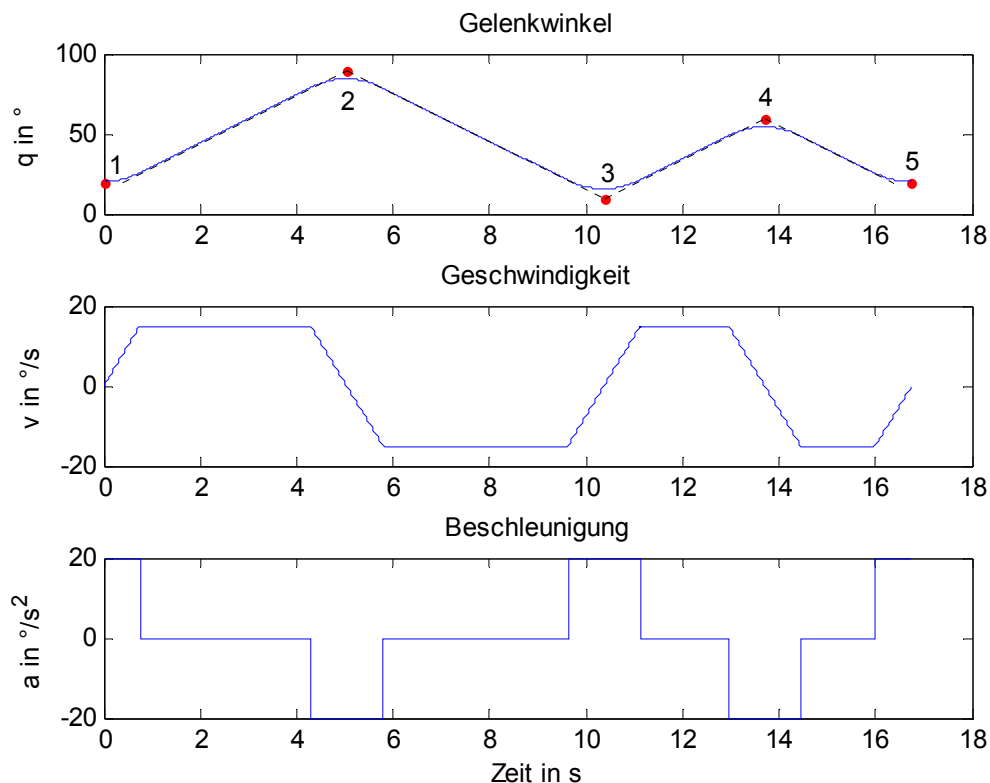


Abbildung 3.4: Verbindung von Via-Punkten durch Geraden mit parabolischen Enden

3.1.3 Anwendung auf mehrere Achsen

Üblicherweise wird bei der Bewegung eines Roboterarms nicht nur eine Achse verändert, sondern mehrere. In diesem Fall muss natürlich für jede Achse der Bewegungsablauf geplant werden. Werden mit dem oben vorgestellten Ansatz der Geraden mit parabolischen Enden mehrere Achsen berechnet, lässt sich in Abbildung 3.5 sehr schnell die Problematik erkennen, die bei einer geforderten synchronen Berechnung von zwei Achsen auftritt. Gerade in dem unten dargestellten Beispiel wird der schlechteste Fall gezeigt. Die erste Achse, also der

blaue Verlauf, hat eine maximale Winkelgeschwindigkeit von $15^\circ/\text{s}$ und eine Beschleunigung von $20^\circ/\text{s}^2$. Die zweite, rot eingezeichnete Achse, erlaubt eine maximale Geschwindigkeit von $20^\circ/\text{s}$ bei einer Beschleunigung von $15^\circ/\text{s}^2$. Das Verhältnis von Beschleunigung zur Geschwindigkeit ist also genau entgegengesetzt. Die Forderung nach Halb-Synchronität setzt voraus, dass beide Achsen die gleiche Zeit benötigen [7]. Dies kann durch Herabskalieren der Geschwindigkeit der zweiten Achse realisiert werden. Wird aber wie in der Aufgabenstellung volle Synchronität gefordert, müssen auch die Phasen der Beschleunigung und die Phasen mit konstanter Geschwindigkeit bei beiden Achsen gleich lange dauern [7]. Im eben vorgestellten Beispiel würde das (bei dem verwendeten Ansatz) die Verfahrdauer beider Achsen maximal verlängern. Es muss nämlich die Geschwindigkeit der schnelleren Achse herunter skaliert werden, während die Achse mit der größeren Beschleunigung diese verkleinern muss. Das heißt, dass bei voller Synchronität beide Achsen mehr Zeit benötigen, als davor die langsamere benötigt hat.

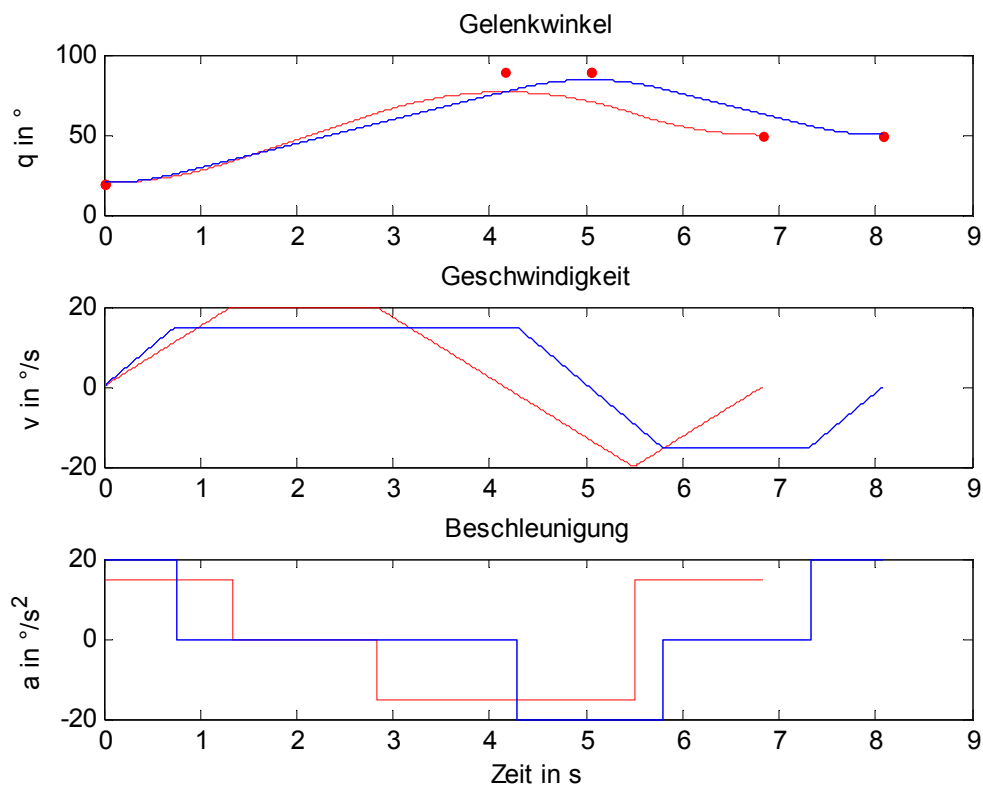


Abbildung 3.5: Zwei Achsen mit unterschiedlichen Grenzwerten

Ein reines Begrenzen der maximalen Geschwindigkeit auf die geringste Geschwindigkeit von allen Achsen sowie die Begrenzung der Beschleunigung auf die geringste Beschleunigung ist dabei nicht zielführend. Im oben dargestellten Beispiel würde dies zwar sehr gut funktionieren. Wenn aber zum Beispiel die langsamste Achse nur eine geringe oder keine Änderung des Gelenkwinkels vornehmen muss, so kann eine schnellere Achse sehr wohl mit ihrer maximalen Geschwindigkeit bewegt werden. Möglich wäre, dass die langsamste Achse immer die gleiche Winkelstellung besitzen soll, während sich eine andere Achse bewegt. In diesem Fall dürfen die schnelleren Achsen nur abgebremst werden, wenn die langsamen Achsen auch tatsächlich den Gelenkwinkel verändern.

3.1.4 Vor- und Nachteile

Bei den Ansätzen von Polynomen 3. Ordnung ist der größte Vorteil, dass die Lösung analytisch sehr einfach berechnet werden kann. Als Nachteil muss zum einen erwähnt werden, dass die Zeiten zwischen den einzelnen gewünschten Gelenkwinkeln vorgegeben werden müssen und sich die Geschwindigkeiten sowie die Beschleunigungen daraus ableiten. Genau diese Eigenschaften sind bei einem Trajektorien-Planer am wenigsten erwünscht. Viel mehr sollte nur anhand der Vorgabe von gewünschten Winkeln und der maximalen Geschwindigkeit und Beschleunigung eine Trajektorie geplant werden, welche die kürzeste Zeit benötigt. Als anderer Nachteil ist die Unvorhersehbarkeit der Trajektorie zu nennen. Bei diesem Ansatz wird zwar sichergestellt, dass die gewünschten Winkelstellungen ohne Abweichung eingenommen werden, auf den Verlauf zwischen diesen Punkten kann so aber kein Einfluss genommen werden. Das Toleranzband würde unter Umständen verlassen werden. Siehe hierzu nachfolgende Abbildung. In diesem Beispiel werden drei Winkel vorgegeben. Diese sind 20° , 290° und 250° . Nach dem Erreichen des zweiten Wertes von 290° nimmt der Betrag der Winkelstellung weiter zu, hier auf $308,2^\circ$ und verringert sich erst später auf den Endwert. Werden von einem übergeordneten Planer diese drei Werte vorgegeben, so ist es nicht erwünscht, dass dazwischen andere Maximalwerte eingenommen werden, zumindest muss aber sichergestellt werden, dass das Toleranzband nicht verlassen wird. Dies kann mit diesem

Ansatz aber nicht garantiert werden. In der Abbildung beträgt die zulässige Toleranz sogar beachtliche $\pm 20^\circ$ und wird trotzdem überschritten. Würden diese Werte auf einen Roboterarm übertragen werden, so wäre es möglich, dass dieser dadurch von der eigentlich geplanten Bahn abweicht und Hindernisse durchfährt.

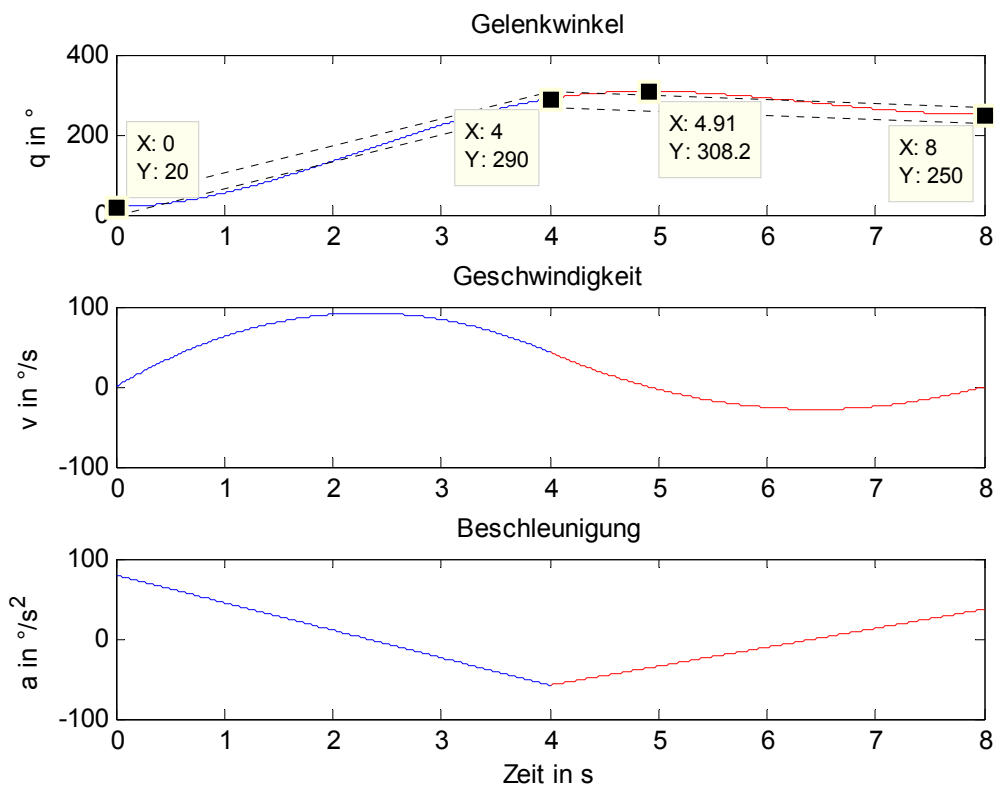


Abbildung 3.6: Abweichung vom Toleranzband bei zwei Polynome dritten Grades

Beim Ansatz der Geraden mit parabolischen Enden bzw. Übergängen ergeben sich mehrere Vorteile. Bereits bei diesem relativ einfachen Ansatz kann die Geschwindigkeit und die Beschleunigung begrenzt werden. Anhand dieser Grenzen wird nun die Zeit ermittelt, die zum Erreichen der vorgegebenen Winkelwerte benötigt wird. Das entspricht bereits den grundlegenden Eigenschaften eines Trajektorienplaners. Trotzdem eignet sich diese Vorgehensweise nicht für die zu untersuchende Aufgabenstellung. Einerseits können so mehrere Achsen nicht bzw. nur schlecht berücksichtigt werden, wie im Abschnitt 3.1.4 beschrieben. Andererseits kann das Erreichen der Via-Punkte nicht garantiert werden. Die Abweichung, dargestellt in Abbildung 3.7, beruht auf dem Abrunden des

Schnittpunktes der Geraden mit einem Parabelbogen. Der Abstand vom Via-Punkt zum Scheitelpunkt der Parabel ist abhängig von der vorgegebenen Maximalbeschleunigung. Bei einer geringeren Beschleunigung wird der Abstand größer und es kann bei Vorgabe einer maximal zulässigen Toleranz nicht garantiert werden, dass diese eingehalten wird. Es wäre zwar denkbar, dass durch Einfügen weiterer Pseudo-Punkte die eigentlichen Via-Punkte getroffen werden [6], zeitoptimal wäre der Planer dadurch allerdings nicht mehr.

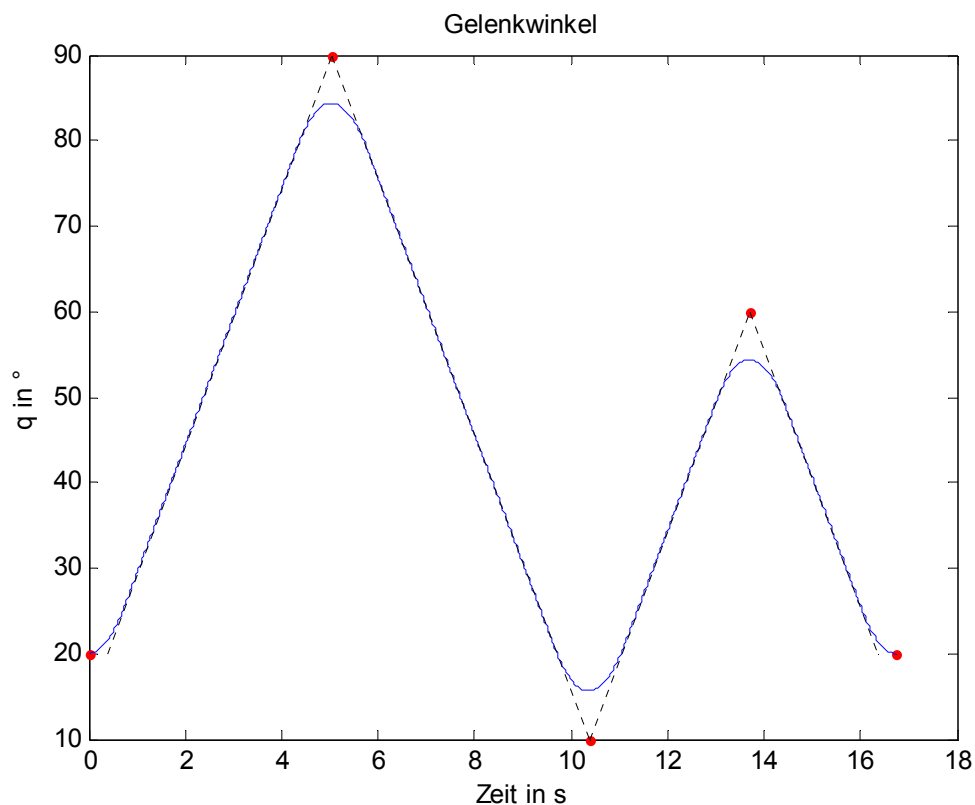


Abbildung 3.7: Abweichung vom Toleranzband bei Geraden mit parabolischen Enden

Außerdem muss die maximale Geschwindigkeit und Beschleunigung in einem einigermaßen sinnvollen Zusammenhang stehen, ansonsten wird der Abstand zu den Via-Punkten viel zu groß und die Trajektorie schwingt nur zwischen den einzelnen Sollwerten hin und her. In Abbildung 3.8 ist die Beschleunigung zu klein um die Maximalgeschwindigkeit erreichen zu können. Durch einen vermischten Ansatz von Vorgabe der Zeitwerte und der Geschwindigkeiten ließe sich dieses Problem zwar eventuell umgehen, allerdings soll die Zeit selbstständig errechnet werden. Ein weiterer Nachteil ist außerdem, dass immer die

maximal mögliche Beschleunigung verwendet wird. Dieses Verhalten kann zu einem erhöhten Verschleiß der Roboterstruktur führen [8].

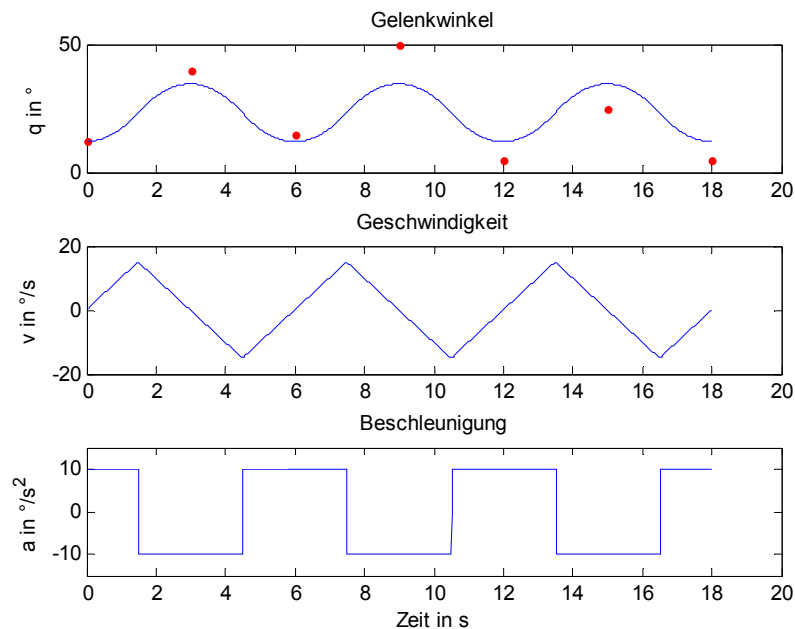


Abbildung 3.8: Abweichung vom Sollverlauf durch unangepasste Werte

Eine kurze Übersicht über alle Vor- und Nachteile der vorgestellten Ansätze, aber auch schon der Ansätze im Kapitel 3.1.6 sowie des Ansatzes im Kapitel 3.2, Trajektoriengenerierung mit nichtlinearem Filter, gibt die nachfolgende Tabelle.

	Ermittlung der Zeit	Analytische Lösung	Abweichung von Via-Punkten	Begrenzung der Maximalwerte	Berücksichtigung mehrerer Achsen
Polynome dritten Grades	Nein	Ja	Nein	Nein	Nein
Gerade parabelförmige Enden	Teils	Ja	Ja	Teils	Nein
Funktionen höherer Ordnung	Nein	Ja	Nein	Nein	Nein
Nichtlineares Filter	Nein	Nein	Ja	Ja	Nein

3.1.5 Mögliche Erweiterungen

In diesem Absatz wird vorgestellt wie bei einer geplanten Trajektorie die Zeit iterativ so verlängert bzw. verkürzt wird, bis die maximalen Werte der Beschleunigung und der Geschwindigkeit erreicht werden [1]. Als Grundlage kann hier der Ansatz von den Polynomen dritten Grades, aber auch die im nächsten Abschnitt vorgestellten Ansätze verwendet werden. Es wird im ersten Schritt eine Trajektorie geplant, die eine bestimmte Verfahrdauer hat. Im nächsten Schritt wird geprüft, ob sich die Werte der Beschleunigung und der Geschwindigkeit innerhalb der festgesetzten Grenzen befinden. Befinden sie sich innerhalb der Grenzen, so wird die Zeit für die Trajektorie um eine bestimmte Schrittweite verkürzt. Überschreiten die Werte bereits die Grenzen, so muss die Zeit um eine festgelegte Schrittweite verlängert werden. Die mit dieser neuen Dauer neu berechnete Trajektorie wird wieder auf die Grenzen geprüft. Dieser Vorgang wird so lange wiederholt, bis die Grenzen erreicht werden, sie dürfen aber nicht überschritten werden. Durch diesen Vorgang kann die kürzeste Zeit für eine Trajektorie ermittelt werden, welche die Grenzen für die maximale Geschwindigkeit und Beschleunigung einhält. In der Abbildung 3.9 links wird eine schrittweise Verkürzung der Zeit dargestellt. In der rechten Abbildung ist die neue Trajektorie mit der kürzesten möglichen Zeit abgebildet. In diesem Beispiel werden als Grenzen für die maximal Winkelgeschwindigkeit $\pm 3^\circ/\text{s}$ und für die maximale Winkelbeschleunigung $\pm 2^\circ/\text{s}^2$ angenommen. Die Zeit wird so lange verkürzt, bis diese Grenzen genau erreicht werden.

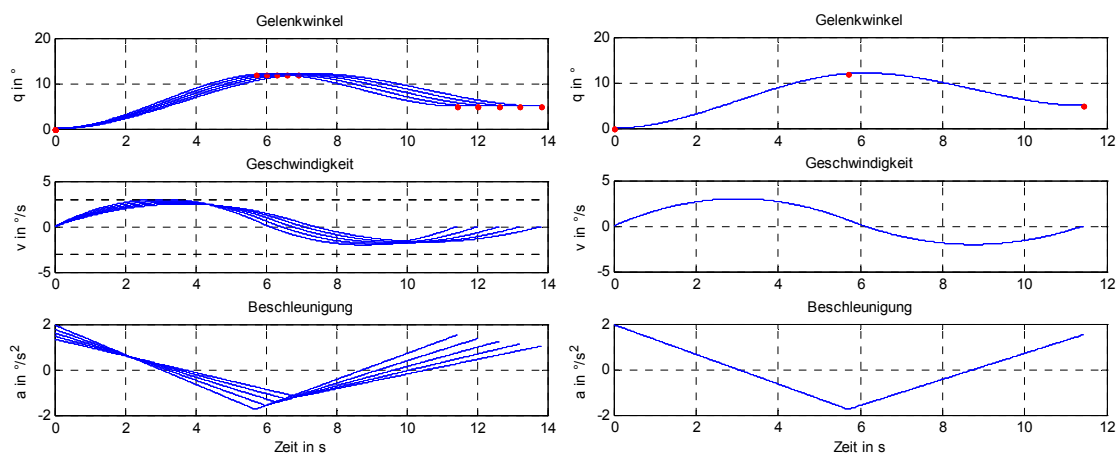


Abbildung 3.9: Iteratives Anpassen an Grenzwerte mit daraus resultierendem Verlauf

3.1.6 Weitere Funktionen

In den vorherigen Abschnitten wurden Polynome dritten Grades und Geraden mit parabolischen Enden vorgestellt. Neben diesen Funktionen gibt es noch eine Vielzahl an weiteren mathematischen Vorgehensweisen, mit denen Punkte im Gelenkwinkelraum verbunden werden können. In diesem Abschnitt wird ein kurzer Überblick über weitere Möglichkeiten gegeben.

Als erstes wird ein Polynom fünften Grades vorgestellt. Dieses Polynom kann sechs Bedingungen erfüllen. Dies wäre der gewünschte Anfangs- und Endwinkel, die Anfangs- und Endgeschwindigkeit sowie die Anfangs- und Endbeschleunigung [1].

Bei einem Polynom siebten Grades kann neben den soeben vorgestellten Bedingungen auch noch der Anfangs- und Endruck vorgegeben werden [1]. Der Ruck stellt die Ableitung der Beschleunigung dar.

Ebenso können auch Polynome noch höheren Grades verwendet werden. Bei diesen drei vorgestellten Berechnungen bleiben aber die Nachteile wie beim Polynom dritten Grades bestehen. Hinzu kommt, dass die maximalen Werte der Ableitungen mit steigendem Grad ansteigen [1].

Eine weitere Möglichkeit, Punkte im Gelenkwinkelraum zu verbinden, bieten trigonometrische Funktionen. Als Vorteil kann dabei angesehen werden, dass für diese Funktionen jede beliebige Ableitung gebildet werden kann. Als Nachteil muss aber gelten, dass mit diesen Funktionen nur zwei Punkte im Gelenkwinkelraum verbunden werden können. Via-Punkte können somit nicht berücksichtigt werden. Außerdem können die Ableitungen am Start- und Endwert einen Sprung aufweisen. Im Folgenden werden trotz der Nachteile drei verschiedene Ansätze von trigonometrischen Funktionen vorgestellt [1].

Dabei ist der erste Winkel q_0 , der zweite ist q_1 .

Harmonische Bewegung:

$$q(t) = \frac{h}{2} \left(1 - \cos \frac{\pi(t-t_0)}{T} \right) + q_0 \quad (3.19)$$

Dabei gilt $h = q_1 - q_0$ und $T = t_1 - t_0$

Zykloidische Trajektorie:

$$q(t) = \left(\frac{t - t_0}{T} - \frac{1}{2\pi} \sin \frac{2\pi(t - t_0)}{T} \right) + q_0 \quad (3.20)$$

Elliptische Trajektorie:

$$q(t) = \frac{h}{2} \left(1 - \frac{\cos \frac{\pi(t - t_0)}{T}}{\sqrt{1 - \alpha \sin^2 \frac{\pi(t - t_0)}{T}}} \right) + q_0 \quad (3.21)$$

Mit $\alpha = (n^2 - 1)/n^2$, n ist das Verhältnis von großer zu kleiner Achse.

Neben dem vorgestellten Ansatz der Gerade mit parabolischen Enden können auch weitere zusammengesetzte Funktionen verwendet werden. Diese sind z. B. Geraden mit kreisförmigen Enden oder auch Geraden mit polynomischen Enden. Der Grad der polynomischen Funktionen kann dabei nahezu beliebig gewählt werden. Anstatt dem geraden Abschnitt in der Mitte kann auch eine Funktion dritten Grades verwendet werden. Die Endstücke werden dabei mit Funktionen vierten Grades berechnet [1].

Diese soeben vorgestellten Ansätze eignen sich aber nur für Punkt zu Punkt Verbindungen und sind deswegen für die gegebene Aufgabenstellung unbrauchbar.

Für die Verbindung von Via-Punkten können weitere, unterschiedliche Ansätze verwendet werden. Um beispielsweise acht Punkte zu verbinden würde aus rein mathematischer Sicht ein Polynom siebten Grades genügen. Müssen nun mehrere Via-Punkte verbunden werden, so steigt auch der Grad der Funktion. Dies kann zu einem starken Oszillieren der Funktion führen. Aus diesem Grund ist dieser Ansatz zu verwerfen.

Ansätze, welche die Via-Punkte nur approximieren, können von vornherein ausgeschlossen werden, da damit die maximal zulässige Toleranz an den einzelnen Punkten nicht mehr garantiert werden kann.

Eine normalerweise sehr beliebte Form verschiedene Punkte miteinander zu verbinden ist die Spline-Funktion. Mit diesem Ansatz kann eine beliebige Anzahl von Punkten verbunden werden. Als Ergebnis werden abschnittsweise de-

finierte Funktionen dritten Grades geliefert, wobei sogar die Krümmung minimiert wird. Wie aber nachfolgend in Abbildung 3.10 dargestellt, kann auch hier der Verlauf nicht vorhergesagt werden. Es ergibt sich derselbe Nachteil wie bei zusammengesetzten Polynomen dritten Grades. Der Verlauf der Funktion nach dem dritten Punkt, hier bei 40° , ist so nicht erwünscht. Die Trajektorie sollte bei diesem Punkt ein lokales Minimum haben und nicht erst danach bzw. die Höhe des neuen lokalen Minimums sollte zumindest vorhersehbar sein. Ein Verlassen des Toleranzbandes ist sehr wahrscheinlich. Lediglich bei sehr großen zulässigen Abweichungen bleibt die Funktion innerhalb des Bandes. Die im Beispiel gewählte Abweichung von $\pm 5^\circ$ wird hier überschritten.

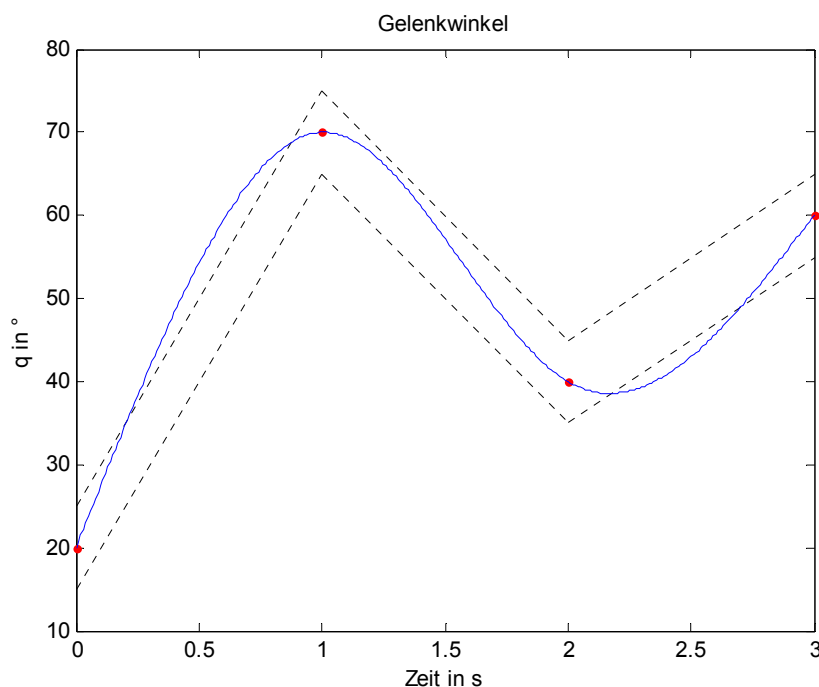


Abbildung 3.10: Abweichung von Spline-Polynome vom Toleranzband

Wird anstelle der Spline-Funktion die B-Spline-Funktion verwendet, ergibt sich dasselbe Problem. Es könnte zwar eventuell durch geschickte Vergabe der Eckpunkte der konvexen Hülle von B-Splines erreicht werden, dass sich diese innerhalb der geforderten Toleranzen befinden, dennoch gestaltet sich dieses Vorgehen als äußerst kompliziert [4].

Weitere sehr interessante Ansätze sind mit Sicherheit Bezier-Kurven oder Nurbs [2]. Hierbei ergibt sich aber auch das Problem, dass die Abweichung an

den einzelnen Punkten nicht immer eingehalten wird. Diese Funktionen folgen zwar ihrem Kontroll-Polygon, sie durchlaufen aber nicht genau die Punkte. Die Abweichung an diesen kann nicht vorhergesagt werden.

3.2 Trajektoriengenerierung mit nichtlinearem Filter

Für die Planung von Trajektorien für Automatisierungsmaschinen mit mehreren Achsen ist die Einhaltung der Beschränkungen der Geschwindigkeit, der Beschleunigung und im besten Fall auch des Rucks von großem Interesse. Normalerweise wird dieses Problem offline berechnet. Ändert sich eine der Bedingungen, muss die komplette Trajektorie neu berechnet werden. Zum Einhalten der Grenzen wird in diesem Abschnitt ein neuartiges Konzept vorgestellt, das dynamische, nichtlineare Filter [1]. Dieses Filter wird in Kaskade zu einem Generator geschaltet, welcher Basis-Bewegungsprofile erzeugt. Diese Profile sind beispielsweise Sprünge oder Rampen. Die Anordnung ist nachfolgend dargestellt. Das Ziel des Filters ist die online Verarbeitung der Bewegungsprofile und sorgt so für eine ausführbare Trajektorie, welche die gegebenen Grenzen einhält.

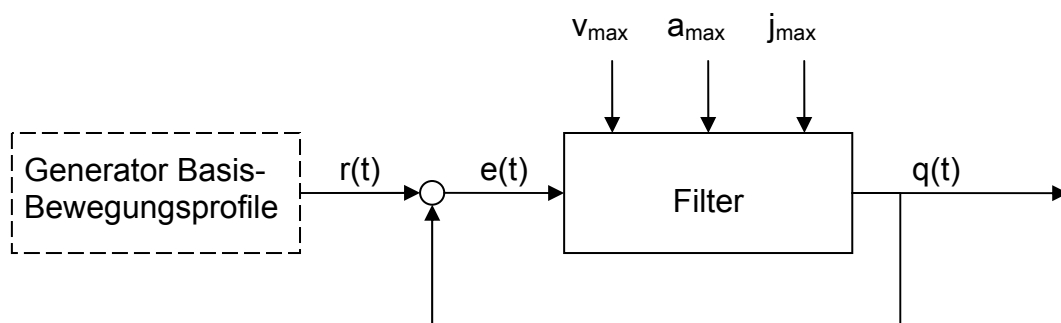


Abbildung 3.11: Anordnung eines nichtlinearen Filters

Das nichtlineare Filter basiert auf einer Rückkopplung und sorgt dafür, dass das Ausgangssignal $q(t)$ dem externen Referenzsignal $r(t)$ folgt und dabei das Einhalten der Grenzen der drei Ableitungen garantiert.

Das Basis-Referenzsignal kann in kürzester Zeit vorgegeben werden. Sollen nun die Beschleunigung, die Geschwindigkeit und der Ruck berücksichtigt werden, so wird ein Filter dritter Ordnung benötigt. In der nächsten Abbildung ist

der Aufbau des Filters dargestellt. Im Anschluss werden nur die für die Implementierung notwendigen Rechenschritte erläutert [1].

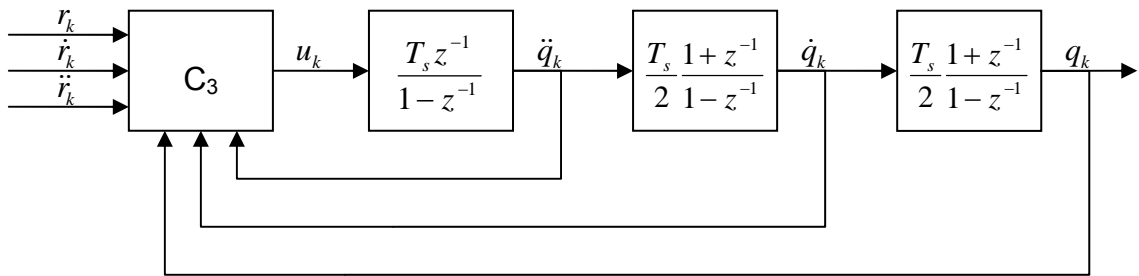


Abbildung 3.12: Aufbau eines nichtlinearen Filters

Das Filter kann in Echtzeit jedes beliebige Standard Referenzsignal $r(t)$ in ein glattes Ausgangssignal $q(t)$ umwandeln und dabei folgende Bedingungen erfüllen:

$$\begin{aligned} v_{\min} &\leq \dot{q}(t) \leq v_{\max} \\ a_{\min} &\leq \ddot{q}(t) \leq a_{\max} \\ -U &= j_{\min} \leq \ddot{q}(t) \leq j_{\max} = +U \end{aligned} \quad (3.22)$$

Der Signal-Generator kann dabei einfache Bewegungsprofile wie Rampen oder Stufen erzeugen. Das Filter kann aber auch anderweitig erzeugte Signale verarbeiten, beispielsweise direkt vom Bediener vorgegebene Bewegungen. Es kann ebenso als Überwachungseinheit eingesetzt werden, welche das von einem eigenständigen Planer erzeugte Ausgangssignal auf Einhalten der Grenzwerte überprüft und nur ausführbare Trajektorien an die Motorregler weiterreicht.

Da dieses Filter digital vorliegt, muss die Zeit diskretisiert werden. Zu jedem Zeitpunkt $t_k = kT_s$ mit $k = 1, 2, \dots$ erhält die Variablenstruktur des Reglers C3 das Referenzsignal r_k und dessen erste und zweite Ableitung, genauso wie die Werte der Position bzw. des Gelenkwinkels q_k , der Geschwindigkeit \dot{q}_k und der Beschleunigung \ddot{q}_k und berechnet den Wert der Ausgangsgröße u_k , was dem Ruck entspricht. Der Ruck muss nun dreimal integriert werden um den Gelenkwinkel zu erhalten. Dazu wird die Beschleunigung durch Rechteck-Approximation integriert, die Geschwindigkeit und die Position mit Trapez-

Approximation. Alternativ kann die Position auch mit der quadratischen Approximation integriert werden.

$$\begin{aligned}\ddot{q}_k &= \ddot{q}_{k-1} + T_s u_{k-1} \\ \dot{q}_k &= \dot{q}_{k-1} + \frac{T_s}{2} (\ddot{q}_k + \ddot{q}_{k-1}) \\ q_k &= q_{k-1} + \frac{T_s}{2} (\dot{q}_k + \dot{q}_{k-1})\end{aligned}\tag{3.23}$$

Der Regler C_3 basiert auf der Regeldifferenz zwischen dem Referenzsignal und dem Wert des Gelenkwinkels q_k , bezogen auf U . Die Variable U ist der maximale Wert der Ausgangs-Variable u_k und repräsentiert somit den Ruck j .

$$e_k = \frac{q_k - r_k}{U}, \quad \dot{e}_k = \frac{\dot{q}_k - \dot{r}_k}{U}, \quad \ddot{e}_k = \frac{\ddot{q}_k - \ddot{r}_k}{U}\tag{3.24}$$

Die Grenzen für die maximale bzw. minimale Geschwindigkeit und Beschleunigung werden auf die Variablen \dot{e}_k und \ddot{e}_k übertragen.

$$\begin{aligned}\dot{e}_{\min} &= \frac{v_{\min} - \dot{r}_k}{U}, & \dot{e}_{\max} &= \frac{v_{\max} - \dot{r}_k}{U} \\ \ddot{e}_{\min} &= \frac{a_{\min} - \ddot{r}_k}{U}, & \ddot{e}_{\max} &= \frac{a_{\max} - \ddot{r}_k}{U}\end{aligned}\tag{3.25}$$

Die Grenzen sind nicht konstant, hängen aber von \dot{r}_k und \ddot{r}_k ab und müssen zu jeder Abtastzeit neu berechnet werden. Andererseits können so während der Laufzeit die Grenzen der Geschwindigkeit und der Beschleunigung verändert werden. Der Regler ändert dann die aktuelle Geschwindigkeit oder Beschleunigung um diese neuen Grenzen wieder einzuhalten.

Die Berechnung des Reglers C_3 ist nachfolgend aufgeführt [1].

$$C_3 : \left\{ \begin{array}{l} \delta = \dot{e}_k + \frac{\ddot{e}_k |\ddot{e}_k|}{2} \\ \sigma = e_k + \dot{e}_k \ddot{e}_k s_\delta - \frac{\ddot{e}_k^3}{6} (1 - 3|s_\delta|) + \frac{s_\delta}{4} \sqrt{2[\ddot{e}_k^2 + 2\dot{e}_k s_\delta]^3} \\ v^+ = e_k - \frac{\ddot{e}_{max}(\ddot{e}_k^2 - 2\dot{e}_k)}{4} - \frac{(\ddot{e}_k^2 - 2\dot{e}_k)^2}{8\ddot{e}_{max}} - \frac{\ddot{e}_k(3\dot{e}_k - \ddot{e}_k^2)}{3} \\ v^- = e_k - \frac{\ddot{e}_{min}(\ddot{e}_k^2 + 2\dot{e}_k)}{4} - \frac{(\ddot{e}_k^2 + 2\dot{e}_k)^2}{8\ddot{e}_{min}} + \frac{\ddot{e}_k(3\dot{e}_k - \ddot{e}_k^2)}{3} \\ \Sigma = \begin{cases} v^+ & \text{wenn } \ddot{e}_k \leq \ddot{e}_{max} \text{ und } \dot{e}_k \leq \frac{\ddot{e}_k^2}{2} - \ddot{e}_{max}^2 \\ v^- & \text{wenn } \ddot{e}_k \geq \ddot{e}_{min} \text{ und } \dot{e}_k \geq \ddot{e}_{min}^2 - \frac{\ddot{e}_k^2}{2} \\ \sigma & \text{ansonsten} \end{cases} \\ u_c = -U \operatorname{sign}(\Sigma + (1 - |\operatorname{sign}(\Sigma)|)[\delta + (1 - |s_\delta|)\ddot{e}_k]) \\ u_k = \max\{u_v(\dot{e}_{min}), \min\{u_c, u_v(\dot{e}_{max})\}\} \end{array} \right. \quad (3.26)$$

Dabei muss s_δ mit $\operatorname{sign}(\delta)$, also der Signum-Funktion ersetzt werden. Diese berechnet sich wie in Formel (3.27) gezeigt.

$$\operatorname{sign}(x) = \begin{cases} +1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases} \quad (3.27)$$

Wird der Ausgangs-Variable u_k die Variable $u_v(v)$ zugewiesen, so erreicht die „Geschwindigkeit“ \dot{e}_k den Wert $\dot{e}_k = v$ in kürzester Zeit. Die Variable $u_v(v)$ wird wie folgt berechnet.

$$C_v : \left\{ \begin{array}{l} u_a(a) = -U \operatorname{sign}(\ddot{e}_k - a) \\ \delta_v(v) = \ddot{e}_k |\ddot{e}_k| + 2(\dot{e}_k - v) \\ u_{cv}(v) = -U \operatorname{sign}(\delta_v(v) + (1 - |\operatorname{sign}(\delta_v(v))|)\ddot{e}_k) \\ u_v(v) = \max\{u_a(\ddot{e}_{min}), \min\{u_{cv}(v), u_a(\ddot{e}_{max})\}\} \end{array} \right. \quad (3.28)$$

Das Ergebnis ist eine Trajektorie, wie sie in Abbildung 3.13 dargestellt wird. Leider hat der Vorgang der Diskretisierung einige Nachteile. Im Besonderen sind das die kleinen Überschwinger der Beschleunigung. Außerdem schwankt der Wert der Ausgangsgröße u_k sehr stark, wenn der Wert null sein sollte. So lange

diese Größe dreimal integriert wird, stellt diese Eigenschaft kein größeres Problem dar.

Nachfolgend werden verschiedene Eingangssignale und unterschiedliche Abtastzeiten T_s dargestellt. In der Abbildung 3.14 sind die Ausgänge des Filters zu sehen, wenn das Referenzsignal eine Stufenform hat. Außerdem wird die maximale Geschwindigkeit während der Trajektorienplanung reduziert, im Beispiel bei ca. 0,73 Sekunden. Bei der linken Abbildung wurde eine Abtastzeit von einer Millisekunde gewählt. Bei der rechten Abbildung beträgt die Abtastzeit 100 Mikrosekunden. Ein sehr großer Unterschied lässt sich bei der Beschleunigung erkennen. Bei geringerer Abtastzeit schwingt diese viel stärker und hat keinen so glatten Verlauf.

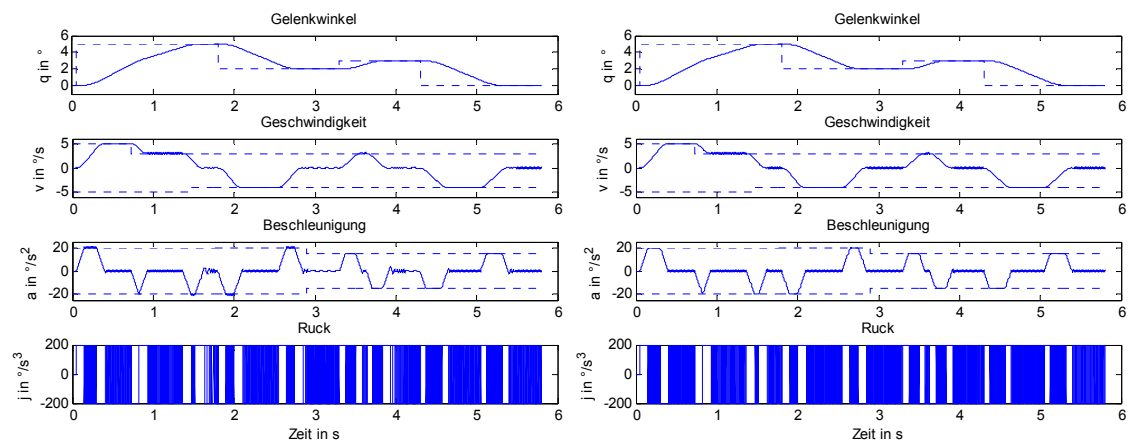


Abbildung 3.13: Verlauf des Ausgangssignals mit unterschiedlichen Abtastzeiten

Als Eingangssignal kann auch ein Sägezahnprofil gewählt werden. Die vom Filter erzeugten Ausgangssignale sind nachfolgend dargestellt. Auch hier beträgt die Abtastzeit in der linken Darstellung eine Millisekunde, in der rechten 100 Mikrosekunden.

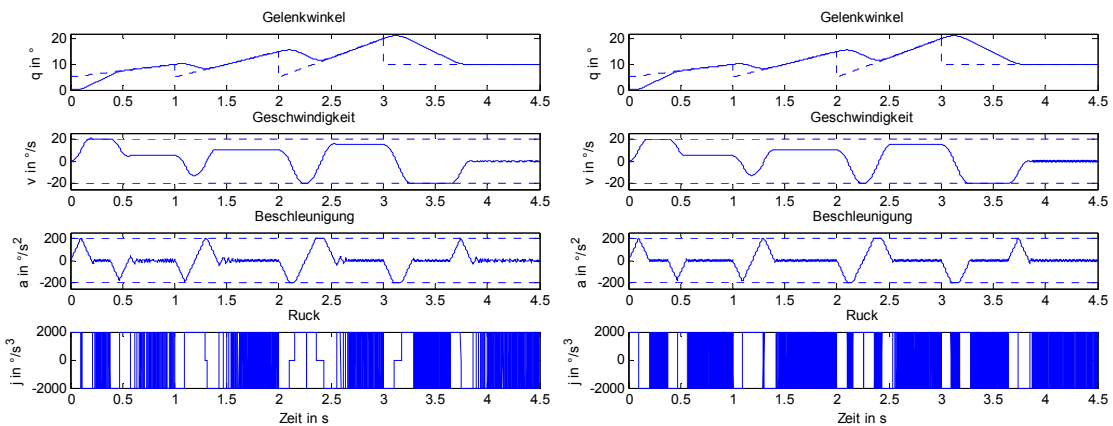


Abbildung 3.14: Ausgang bei einem Sägezahnprofil mit unterschiedlichen Abtastzeiten

Wird nun aber schon eine geplante Trajektorie durch das Filter verarbeitet, so erhält man eine Trajektorie, die genau dem Eingangssignal folgt, wenn dabei die Grenzwerte nicht überschritten werden. Ansonsten hat die Einhaltung der Grenzen absolute Priorität.

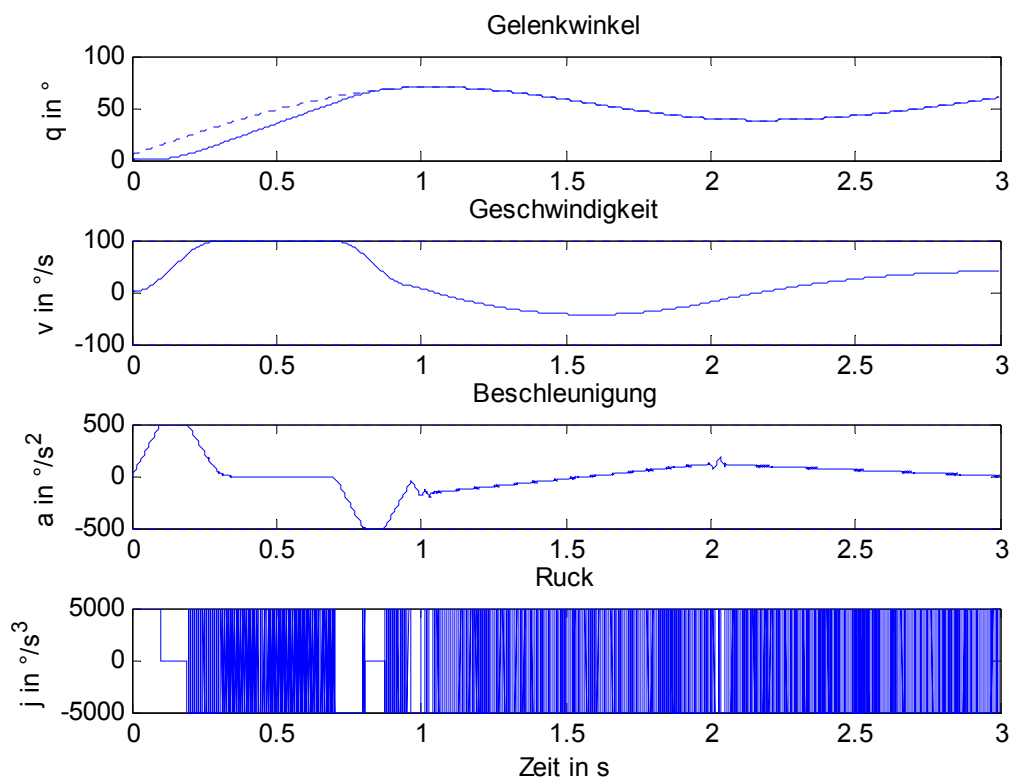


Abbildung 3.15: Ausgang des Filters mit stetigem Eingangssignal

Der Hauptverwendungszweck für diese Trajektorienplanung ist das Einfangen eines unbekannten Signals. Als typisches Beispiel kann die Bearbeitung von

Werkstücken auf einem Förderband angesehen werden. Der Roboter bzw. die Maschine mit dem Bearbeitungswerkzeug muss sich genau über das Werkstück positionieren. Die Geschwindigkeit des Förderbandes ist dabei bekannt und konstant. Das Werkzeug muss auf die Bearbeitungsstücke warten und diese dann „einfangen“, sich also mit der Geschwindigkeit des Objekts synchronisieren. Mit einer Offline-Planung wäre die Lösung dieses Problems sehr komplex, sehr viel einfacher kann es durch die Benutzung des nichtlinearen Filters gelöst werden [1].

3.3 Optimale Trajektorienplanung

Wird wie im nächsten Kapitel beschrieben die Pfadplanung von der Trajektorienplanung getrennt, so muss die x-Achse neu skaliert werden. Bei der entkoppelten Trajektorienplanung wird so nur der vorher festgelegte Pfad zeitoptimal [5]. Andere, eventuell bessere, Pfade werden dabei aber nicht mehr berücksichtigt. Soll direkt im Zustandsraum eine zeitoptimale Trajektorie geplant werden, so kann dazu die so genannte „optimal control theory“ verwendet werden. Die dazu erforderlichen Bedingungen sind für nahezu jedes Robotermodell sehr komplex und können nicht analytisch gelöst werden. Zwei numerische Näherungen sind zum Beispiel die nichtlineare Optimierung und die „grid-based search“, also die gitterbasierte Suche.

Das Ergebnis mit der nichtlinearen Optimierung ist der entkoppelten Trajektorienplanung sehr ähnlich, benötigt aber im Initialzustand Schätzungen, von denen die Lösung stark abhängig ist [5]. Aus diesem Grund wird der entkoppelten Trajektorienplanung der Vorzug gegeben.

Bei der gitterbasierten Suche kann vom Benutzer vorgegeben werden, wie nah zum Zeitoptimalen die Bewegung sein soll. Allerdings wird der Endpunkt nicht genau erreicht. Vor allem die stark anwachsende Größe des Gitternetzes ist ein großer Nachteil [5]. Dieser Ansatz eignet sich besser für die Planung von Trajektorien, die z. B. ein autonomer mobiler Roboter beim Umfahren von Hindernissen einnehmen soll.

4 Entkoppelte Trajektorienplanung

Bei der entkoppelten Trajektorienplanung sind die Pfadplanung und die Trajektorienplanung getrennt [1][3][5][8]. Wie in der Abbildung 4.1 zu erkennen ist wird zunächst ein Pfad $q(s)$ erzeugt. Dieser Pfad wird nicht über die Zeit t , sondern über den Bahnparameter s berechnet. Um nun eine Trajektorie zu erhalten, muss dieser Parameter im Zustandsraum neu skaliert werden. Bei der Skalierung ist die Berücksichtigung der Aktor-Grenzen von großer Bedeutung, somit ergibt sich ein $s(t)$. Wird dies nun auf den Pfad angewandt, so ergibt sich $q(s(t))$. Nun ist q abhängig von der Zeit, es ergibt sich also eine Trajektorie $q(t)$. In diesem Kapitel wird die Vorgehensweise für diesen Ansatz beschrieben.

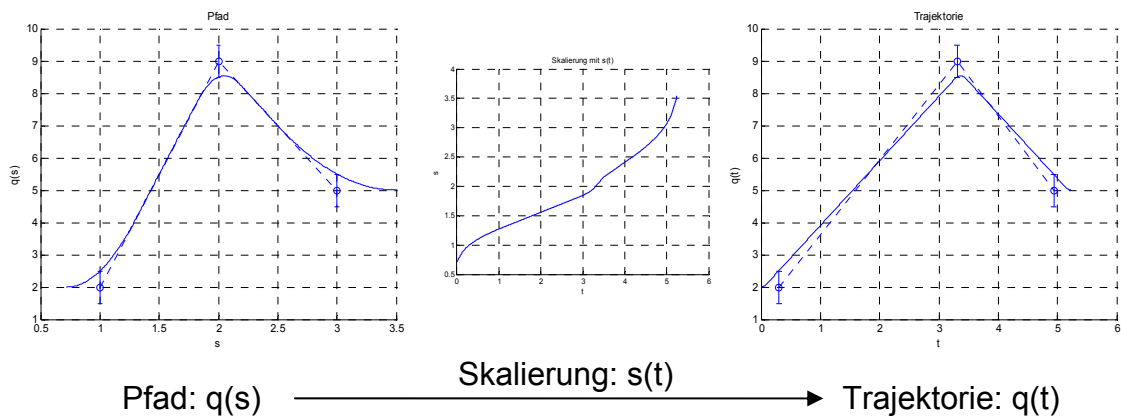


Abbildung 4.1: Erzeugung von Trajektorien durch Skalierung des Pfades

4.1 Eigenschaften

Der Algorithmus soll Trajektorien für vom Planer vorgegebene Gelenkwinkel erzeugen. Sollen nun Via-Punkte ohne Stillstand durchfahren werden, kann hier mit abschnittswiseen Funktionen, ähnlich Spline-Funktionen, gerechnet werden. Die Trajektorienplanung sorgt dafür, dass das Leistungspotential eines Roboters voll ausgenutzt werden kann. Dabei soll die Geschwindigkeit entlang der vorher geplanten Bahn möglichst hoch sein um, damit die Verfahrdauer bzw. bei Industrierobotern die Taktzeit zu verkürzen. Außerdem sollen bei der Trajek-

torienplanung weiche Bewegungen erzeugt werden. Diese sollen einen hohen Verschleiß der Roboterstruktur verhindern [8].

In der Literatur gibt es verschiedene Definitionen von zeitoptimalen Trajektorien. In dieser Arbeit wird die gängige Definition verwendet, dass eine Trajektorie zeitoptimal ist, wenn die Verfahrdauer des Roboters auf einer vorher festgelegten Bahn minimiert wird [8]. Dabei wird allerdings nur diese Bahn zeitoptimal, andere Bahnen, welche ebenso die gewünschten Zielpunkte im Raum treffen, werden hiermit nicht mehr berücksichtigt. Es könnten also noch bessere Trajektorien ermittelt werden, die mit diesem Ansatz entworfene Trajektorie ist aber auf den gesamten Raum betrachtet schon nahezu zeitoptimal. Eventuell würden mit der im Abschnitt 3.3 erwähnten Vorgehensweise geringfügig bessere Trajektorien entworfen, was sich allerdings stark auf die Rechenzeit auswirkt. Die entkoppelte Trajektorienplanung verspricht hier den besten Kompromiss.

4.2 Pfaderzeugung

Für die Erzeugung des Pfades sind nahezu alle vorher vorgestellten Ansätze denkbar. Auch hier dienen als Grundlage die Sollwinkel q , die von einem übergeordneten Planer erzeugt werden. Auch diese Winkelwerte können wieder in ein Koordinatensystem eingetragen werden, siehe hierzu Abbildung 4.2. Anders als bei der Trajektorienplanung im Kapitel 3.1 wird hier auf der x -Achse nicht die Zeit, sondern der Pfadparameter s aufgetragen. Beispielsweise kann jede ganze natürliche Zahl des Parameters einem Zustand entsprechen, also einer Winkelstellung, die vom Planer gefordert ist. Die Sollwinkel für die Startwerte jeder Achse werden alle z. B. über den Parameter $s = 1$ aufgetragen. In der Abbildung sind diese Winkel durch rote Punkte dargestellt. Die Sollwerte für den nächsten Zustand müssen bei $s = 2$ eingetragen werden, der nächste Zustand bei $s = 3$ und so weiter. Denkbar ist aber auch, dass die Startwerte bei $s = 0$ beginnen oder dass der letzte Wert bei $s = 1$ steht. Eigentlich sind hier viele Variationen möglich, da hier die x -Achse nicht den zeitlichen Verlauf darstellt.

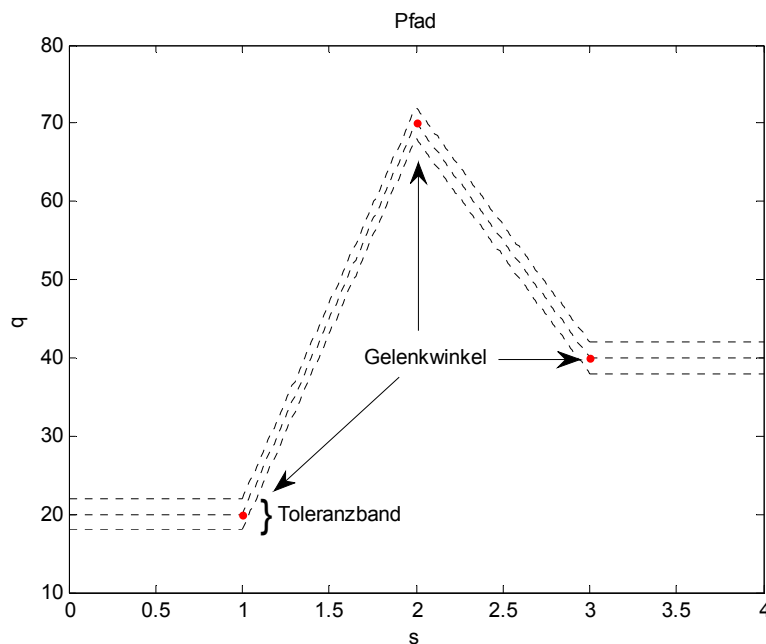


Abbildung 4.2: Vorgabe des Verlaufs für die Pfadplanung

Durch Auftragen des Winkels jeder Achse über den gleichen s -Parameter kann bereits ein halb-synchroner Ablauf der Bewegung erzeugt werden. Nach der Skalierung befindet sich dann nämlich ein jeder Sollwinkel über dem gleichen Zeitpunkt. Wie eine voll-synchrone Bewegung erzeugt wird, ist im Abschnitt 5.1 beschrieben.

Bei der Erzeugung von $q(s)$ ist zu beachten, dass die mathematische Funktion, die durch die einzelnen Punkte gelegt wird, glatt ist. Außerdem muss sie mindestens zweimal ableitbar sein, für Berücksichtigung des Rucks sogar dreimal. Die später ermittelte Trajektorie kann nämlich nicht glatter sein als der jetzt erstellte Pfad. Anders als bei der Trajektorienplanung entspricht hier die erste Ableitung der Pfadgeschwindigkeit und nicht der Gelenkwinkelgeschwindigkeit. Das heißt dass hier nicht die Grenzwerte der Antriebe berücksichtigt werden müssen. Auch die zweite Ableitung entspricht nicht der Beschleunigung des Motors, sondern der Pfadbeschleunigung. Dadurch sind bei der Pfadplanung viel mehr Möglichkeiten zum Verbinden der einzelnen Winkelstellungen vorhanden. Es könnten beispielsweise Radien gewählt werden, die eine sehr große Krümmung aufweisen. Bei der Pfadplanung soll das ganze Augenmerk auf die korrekte Einnahme der Winkelstellungen liegen. Hier muss die Toleranz in

den Via-Punkten ebenso berücksichtigt werden wie auch das Verhindern von Überschwingern, die zum Beispiel von Polynomen dritten Grades erzeugt werden. Dazu wird die Verbindung von zwei Punkten durch ein Toleranzband dargestellt. Somit kann sowohl die Einhaltung der Toleranz im Via-Punkt als auch das Verhindern von Überschwingern garantiert werden. Es kann nun eine beliebige, eventuell auch zusammengesetzte mathematische Funktion ermittelt werden, welche die Via-Punkte verbindet und dieses Toleranzband nicht verlässt. Als gut zu realisierende Methode haben sich zusammengesetzte Funktionen erwiesen. Hier kann jeder Abschnitt zwischen zwei Via-Punkten separat betrachtet werden, es muss aber auf einen stetigen Übergang zur Nachbarfunktion geachtet werden. In dem unten dargestellten Beispiel sind für zwei Achsen die Via-Punkte über die ganzen Zahlen der x-Achse aufgetragen.

Ein möglicher Pfad für zwei Achsen ist nachfolgend dargestellt. Hierbei ist sehr schön zu erkennen, dass der Pfad das Toleranzband genau einhält. Der Pfad, der in das Toleranzband gelegt wird, beginnt hier allerdings nicht bei $s = 0$. Deswegen wird der gesamte Pfad in der rechten Abbildung so weit nach links verschoben, bis er beim Nullpunkt startet.

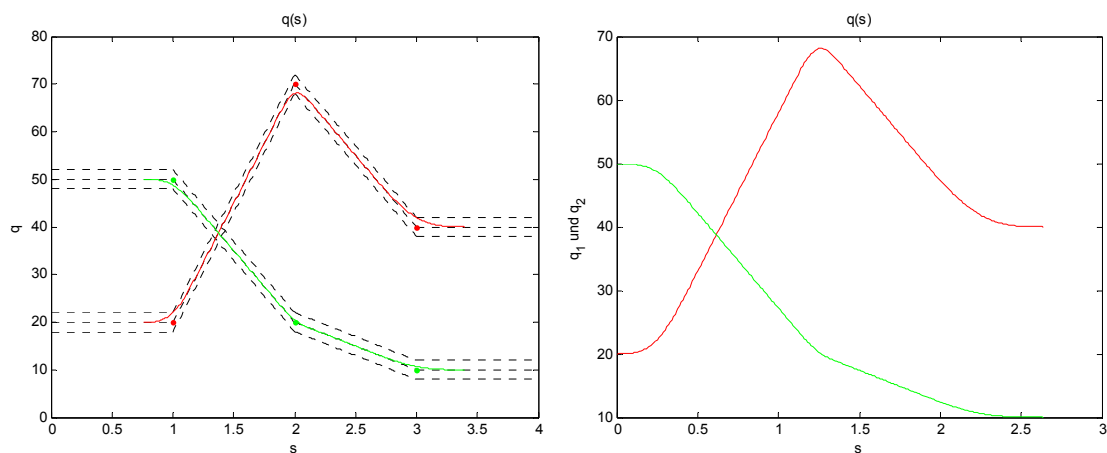


Abbildung 4.3: Geplanter Pfad im Toleranzband und mit Start von $s = 0$

Aus welchen mathematischen Funktionen sich der Pfad zusammensetzt, wird im nächsten Kapitel „Umsetzung in einen Algorithmus“ genauer beschrieben. Für die weitere Betrachtung der Berechnung des zeitlichen Verlaufs ist es notwendig, ein Beispiel eines möglichen Verlaufs des Pfades darzustellen. Der

Pfad sowie dessen erste und zweite Ableitung sind in der nächsten Abbildung dargestellt. Das linke Bild zeigt dabei den über den Parameter s aufgetragen geplanten Pfad. Im rechten Bild sind die erste und die zweite Ableitung zu sehen. Diese Ableitungen entsprechen der Pfadsteigung bzw. der Pfadkrümmung.

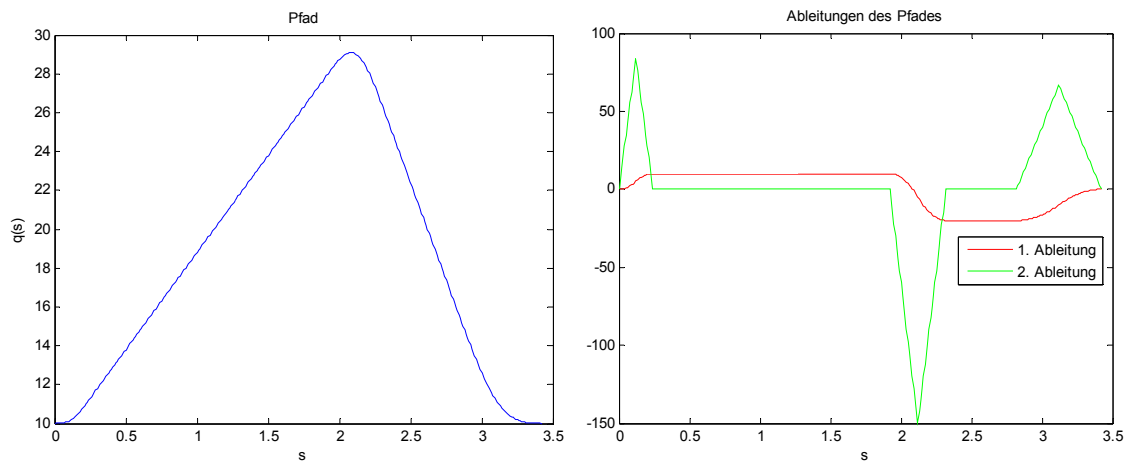


Abbildung 4.4: Erzeugter Pfad mit dazugehörigen Ableitungen

4.3 Berechnung des zeitlichen Verlaufs

Wie eingangs in diesem Kapitel schon erwähnt wurde, muss die x-Achse neu skaliert werden um vom Bahnparameter s auf die Zeit t zu kommen. Für diese Skalierung gibt es verschiedene Möglichkeiten, die im Anschluss vorgestellt werden.

Als Bedingung, dass aus einem stetigen Pfad $q(s)$ eine stetige Trajektorie $q(t)$ erzeugt wird, muss gelten, dass $s(t)$ auch stetig ist.

$$q = q(s(t)) \quad (4.1)$$

Dasselbe gilt für die Ableitungen des Pfades nach der Zeit t . Diese berechnen sich wie folgt [1][5]:

$$\dot{q} = \frac{dq(s)}{dt} = \frac{dq(s)}{ds} \frac{ds}{dt} = \frac{dq(s)}{ds} \dot{s} \quad (4.2)$$

Für die zweite Ableitung muss gelten:

$$\ddot{q} = \frac{d\dot{q}}{dt} = \frac{d^2q(s)}{ds^2} \dot{s}^2 + \frac{dq(s)}{ds} \ddot{s} \quad (4.3)$$

Und schließlich für die dritte Ableitung:

$$\ddot{q} = \frac{d^3 q(s)}{ds^3} \dot{s}^3 + \frac{d^2 q(s)}{ds^2} 3 \dot{s} \ddot{s} + \frac{dq(s)}{ds} \ddot{\ddot{s}} \quad (4.4)$$

Damit die erste und die zweite Ableitung der Trajektorie stetig sind, müssen beide Ableitungen des Pfades und beide Ableitungen von $s(t)$ stetig sein. Die dritte Ableitung muss zwar nicht stetig, aber bei Berücksichtigung des maximal möglichen Rucks zumindest begrenzt sein. Das heißt, dass die zweiten Ableitungen von $q(s)$ und von $s(t)$ noch einmal ableitbar sein müssen.

4.3.1 Konstante Skalierung

Wie die Bezeichnung schon beschreibt, wird hier mit einem konstanten Faktor skaliert [1]. Ziel der Skalierung ist es, aus dem Pfad eine Trajektorie zu erzeugen, die alle Grenzen der Achsen einhält. Nach der Skalierung repräsentiert die x-Achse die Zeit, und die erste und zweite Ableitung stehen für die tatsächlich vom Antrieb geforderten Werte. Die erste Ableitung steht jetzt für die Winkelgeschwindigkeit, also die Geschwindigkeit, die der Motor aufbringen muss um eine Änderung des Gelenkwinkels herbei zu führen. Die zweite Ableitung beschreibt nach der Skalierung die Beschleunigung, die der Motor zum Einnehmen einer neuen Winkelgeschwindigkeit ausführen muss. Diese Werte müssen nun aber so begrenzt werden, dass die vom Motor maximal zur Verfügung gestellte Geschwindigkeit und Beschleunigung nicht überschritten werden. Zieht man das in Abschnitt 4.2 vorgestellte Beispiel zu Rate und nimmt als Obergrenze für die Winkelgeschwindigkeit $10^\circ/\text{s}$ und für die Winkelbeschleunigung $20^\circ/\text{s}^2$ an, muss die x-Achse so neu parametrisiert werden, damit die beiden Ableitungen diese Grenzen nicht überschreiten.

Die Skalierung $s(t)$ mit dem Skalierungsfaktor λ ist im Anschluss dargestellt [1].

$$\begin{aligned} s(t) &= \lambda t \\ \dot{s} &= \lambda \\ \ddot{s} &= 0 \\ &\vdots \end{aligned} \quad (4.5)$$

Eingesetzt in die Formeln (4.1) bis (4.4) ergibt sich somit:

$$\begin{aligned}
 q(t) &= q(s(t)) \\
 \dot{q}(t) &= \frac{dq(s(t))}{ds} \lambda \\
 \ddot{q}(t) &= \frac{d^2 q(s(t))}{ds^2} \lambda^2 \\
 \dddot{q}(t) &= \frac{d^3 q(s(t))}{ds^3} \lambda^3 \\
 &\vdots
 \end{aligned} \tag{4.6}$$

Ein Skalierungsfaktor λ , der die maximale Gelenkwinkelgeschwindigkeit \dot{q} einhält, kann durch Umstellung der zweiten Formel in (4.6) erreicht werden.

$$\lambda = \frac{\dot{q}_{max}}{\left| \frac{dq}{ds} \right|_{max}} \tag{4.7}$$

Sollen mit dem Skalierungsfaktor sowohl die Gelenkwinkelgeschwindigkeit als auch die Beschleunigung und der Ruck berücksichtigt werden, so wird λ folgendermaßen berechnet.

$$\lambda = \min \left\{ \frac{\dot{q}_{max}}{\left| \frac{dq(s)}{ds} \right|_{max}}, \sqrt{\frac{\ddot{q}_{max}}{\left| \frac{d^2 q(s)}{ds^2} \right|_{max}}}, \sqrt[3]{\frac{\dddot{q}_{max}}{\left| \frac{d^3 q(s)}{ds^3} \right|_{max}}} \right\} \tag{4.8}$$

Bei mehreren Achsen wird für jede Achse ein λ berechnet, und von all diesen das Kleinste genommen.

Die Umrechnung des Pfades mit $s(t)$ und konstantem Skalierungsfaktor ist nachfolgend, die erste und die zweite Ableitung ist in Abbildung 4.6 dargestellt.

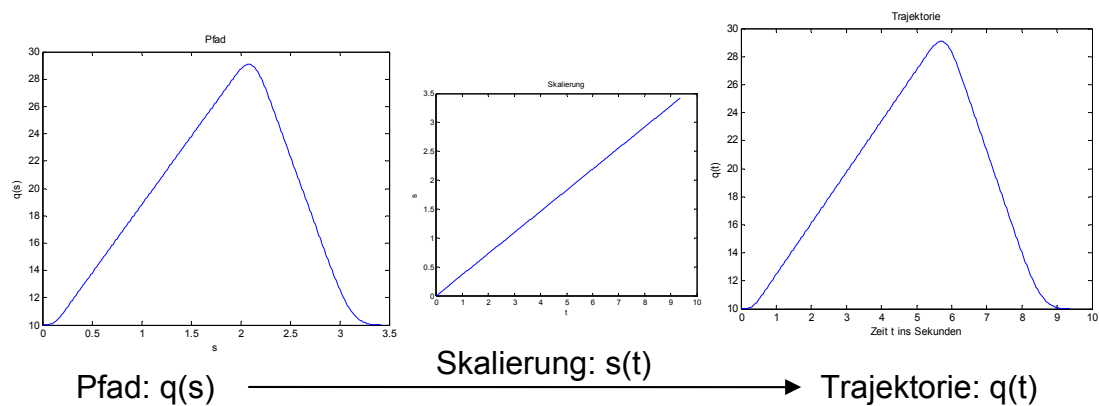


Abbildung 4.5: Erzeugung von Trajektorien durch konstante Skalierung des Pfades

In der obigen Abbildung lässt sich erkennen, dass ein konstanter Skalierungsfaktor vorliegt, das Verhältnis von der Zeit zum Pfadparameter s ergibt eine Gerade. Die Trajektorie verläuft genau so wie der Pfad, lediglich die Skalierung der x-Achse hat sich geändert. In nachfolgender Abbildung sind die erste und die zweite Ableitung dargestellt. Diese beiden Ableitungen repräsentieren die tatsächliche Gelenkwinkelgeschwindigkeit bzw. -beschleunigung. Der Nachteil bei konstanter Skalierung ist, dass der Skalierungsfaktor von der größten Abweichung der Ableitungen des Pfades zu den dazugehörigen Grenzen bestimmt wird, in diesem Beispiel wäre das die Pfadbeschleunigung bei $s = 2,12$, wie im rechten Bild in Abbildung 4.4 zu erkennen ist. Dieses Maximum muss so weit herunter skaliert werden, bis die Grenze von $20 \text{ }^\circ/\text{s}^2$ eingehalten wird. In Abbildung 4.6 werden zwar alle Grenzen eingehalten, jedoch ist noch Verbesserungspotenzial vorhanden, wenn es um die Geschwindigkeit geht. Besonders im Bereich von $t = 0,65$ bis $t = 5,26$ sowie im Bereich von $t = 6,35$ bis $t = 7,72$ wird die maximal mögliche Geschwindigkeit des Antriebs von $10 \text{ }^\circ/\text{s}$ nicht ausgenutzt.

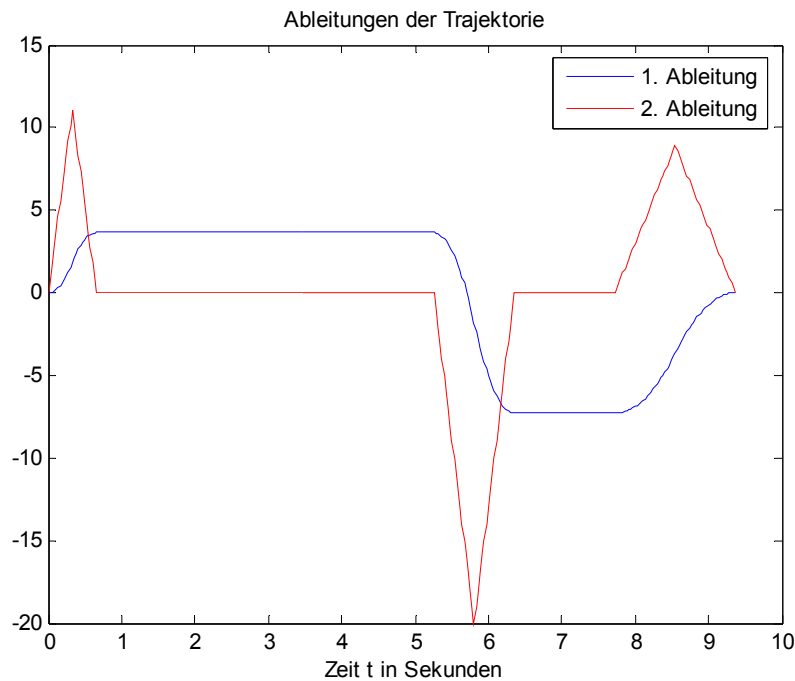


Abbildung 4.6: Ableitungen der Trajektorie bei konstanter Skalierung

Ein nahe liegender Gedanke die Grenzwerte über den gesamten Verlauf voll auszunutzen wäre, die Skalierung nicht über den gesamten Pfad, sondern abschnittsweise festzulegen. Die Skalierung λ ist also nun nicht mehr konstant, sondern wird zu einem $\lambda(s)$. Wird die Skalierung λ für jedes $\Delta s = 0,01$ festgelegt, so ergibt sich der nachfolgend dargestellte, nicht stetig differenzierbare Verlauf der Skalierung. Möglich ist auch, dass das λ nun nicht mehr stetig ist.

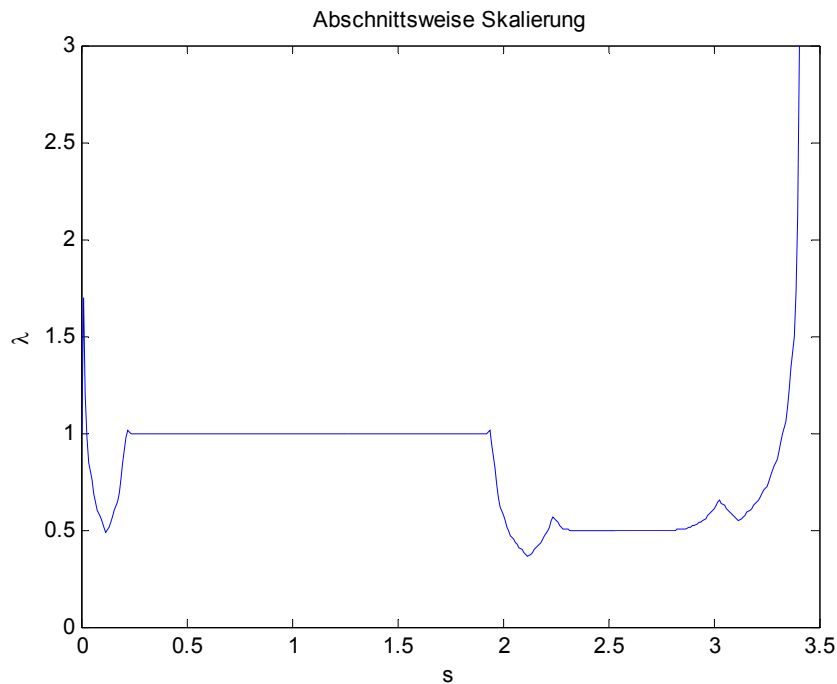


Abbildung 4.7: Verlauf des abschnittsweise festgelegten λ

Mit der dargestellten Skalierung ist nun auch $s(t)$ nicht mehr stetig differenzierbar. Der Verlauf von $s(t)$ und die zur neu ermittelten Trajektorie gehörenden Ableitungen sind in Abbildung 4.8 dargestellt.

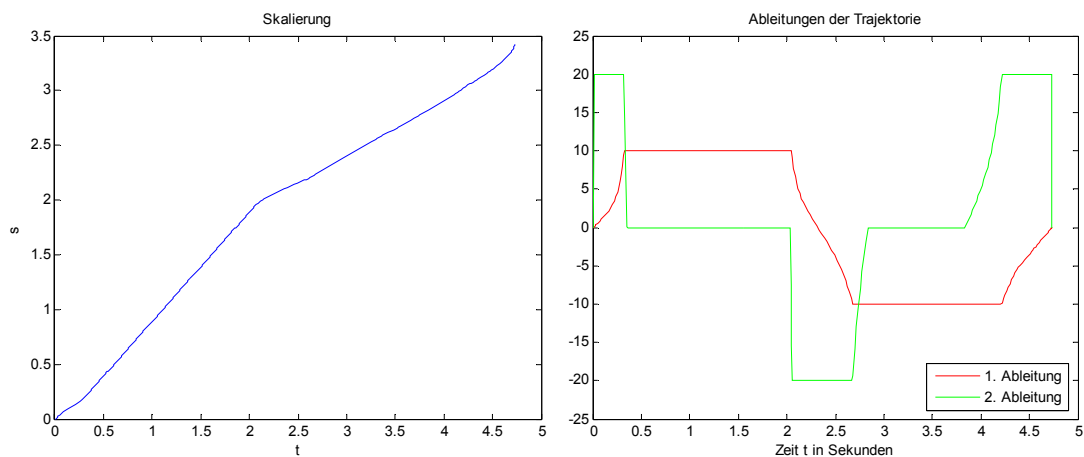


Abbildung 4.8: Abschnittsweise Skalierung mit fehlerhaften Ableitungen der Trajektorie

Durch dieses neue $s(t)$ wird auch $q(t)$ nicht stetig differenzierbar und somit alle zeitlichen Ableitungen von $q(t)$ unstetig. In der rechts gezeigten Darstellung der Ableitungen werden zwar die festgelegten Grenzen eingehalten und gut genutzt, jedoch sind die Unstetigkeiten deutlich zu erkennen. Die Beschleunigung

weist nun Sprünge auf. Bei näherer Betrachtung der beiden Ableitungen lässt sich außerdem erkennen, dass diese nicht mehr zueinander passen.

Ein Scheitern dieses Vorgehens lässt sich formell verdeutlichen. Die in (4.5) sehr einfache Skalierung und deren 1. Ableitung werden mit einem $\lambda(s)$ zu:

$$\begin{aligned} s(t) &= \lambda(s) t \\ \dot{s}(t) &= \lambda(s) + \frac{d\lambda(s)}{ds} \dot{s} t \\ \dot{s}(t) &= \frac{\lambda(s)}{1 - \frac{d\lambda(s)}{ds} t} \end{aligned} \quad (4.9)$$

Für eine Berücksichtigung der maximalen Gelenkwinkelgeschwindigkeit muss dieses \dot{s} in (4.2) eingefügt werden.

$$\dot{q} = \frac{dq}{ds} \left(\frac{\lambda(s)}{1 - \frac{d\lambda(s)}{ds} t} \right) \quad (4.10)$$

Ein einfaches Auflösen nach $\lambda(s)$ wie in Formel (4.7) ist so nicht mehr möglich. Jetzt müsste dazu eine Differentialgleichung gelöst werden.

Um nun eine Skalierung zu erhalten, welche die Grenzen des Antriebs berücksichtigt, aber auch gut ausnutzt, keine Unstetigkeiten und keine ungültigen Ableitungen erzeugt, wird der Skalierungsfaktor durch numerische Integration ermittelt. Dieses Vorgehen wird nachfolgend beschrieben.

4.3.2 Variable Skalierung durch Integration

Um Unstetigkeiten zu vermeiden ist ein anderes Vorgehen nötig. Als erstes müssen die Gleichungen (4.2) und (4.3), so wie anschließend gezeigt, umgestellt werden [5].

$$\dot{s} = \frac{\dot{q}}{\frac{dq(s)}{ds}} \quad (4.11)$$

$$\ddot{s} = \frac{\ddot{q} - \frac{d^2q(s)}{ds^2} \dot{s}^2}{\frac{dq(s)}{ds}} \quad (4.12)$$

In diesen Gleichungen können nun die Grenzen des Antriebs berücksichtigt werden. Dabei entspricht \dot{q} der Gelenkwinkelgeschwindigkeit und \ddot{q} der Beschleunigung. Mit Einsetzen der maximal und minimal möglichen Grenzen ergeben sich so folgende Ausdrücke.

$$\dot{s}_{\max}(s) = \begin{cases} \frac{\dot{q}_{\max}}{\frac{dq(s)}{ds}}, & \frac{dq(s)}{ds} > 0 \\ \frac{\dot{q}_{\min}}{\frac{dq(s)}{ds}}, & \frac{dq(s)}{ds} < 0 \end{cases} \quad \text{mit } \dot{q}_{\min} \leq \dot{q} \leq \dot{q}_{\max} \quad (4.13)$$

$$\dot{s}_{\min}(s) = \begin{cases} \frac{\dot{q}_{\min}}{\frac{dq(s)}{ds}}, & \frac{dq(s)}{ds} > 0 \\ \frac{\dot{q}_{\max}}{\frac{dq(s)}{ds}}, & \frac{dq(s)}{ds} < 0 \end{cases} \quad \text{mit } \dot{q}_{\min} \leq \dot{q} \leq \dot{q}_{\max} \quad (4.14)$$

Falls $\frac{dq(s)}{ds} = 0$ ist, wird das Ergebnis unendlich. Selbiges gilt für die folgenden Gleichungen:

$$\ddot{s}_{\max}(s, \dot{s}) = \begin{cases} \frac{\ddot{q}_{\max} - \frac{d^2q(s)}{ds^2} \dot{s}^2}{\frac{dq(s)}{ds}}, & \frac{dq(s)}{ds} > 0 \\ \frac{\ddot{q}_{\min} - \frac{d^2q(s)}{ds^2} \dot{s}^2}{\frac{dq(s)}{ds}}, & \frac{dq(s)}{ds} < 0 \end{cases} \quad \text{mit } \ddot{q}_{\min} \leq \ddot{q} \leq \ddot{q}_{\max} \quad (4.15)$$

$$\ddot{s}_{\min}(s, \dot{s}) = \begin{cases} \frac{\ddot{q}_{\min} - \frac{d^2q(s)}{ds^2} \dot{s}^2}{\frac{dq(s)}{ds}}, & \frac{dq(s)}{ds} > 0 \\ \frac{\ddot{q}_{\max} - \frac{d^2q(s)}{ds^2} \dot{s}^2}{\frac{dq(s)}{ds}}, & \frac{dq(s)}{ds} < 0 \end{cases} \quad \text{mit } \ddot{q}_{\min} \leq \ddot{q} \leq \ddot{q}_{\max} \quad (4.16)$$

Eine Möglichkeit, konsistentes $s(t)$ zu erhalten, besteht aus zweifacher Integration von \ddot{s} unter Berücksichtigung der gezeigten Grenzen von \dot{s} und \ddot{s} [5][8]. Zunächst muss die maximal mögliche Pfadgeschwindigkeit, also \dot{s}_{\max} in den Zustandsraum aufgetragen werden. In diesem Zustandsraum wird der Parameter s auf der x-Achse, die zeitliche Ableitung davon, also \dot{s} , auf der y-Achse aufgetragen. Durch Einzeichnen der maximalen Pfadgeschwindigkeit \dot{s}_{\max} ergibt sich die Obergrenze für \dot{s} . Diese Grenze ist in der nachfolgenden Abbildung

durch die dicke schwarze Linie gekennzeichnet. Diese wird auch Maximale-Geschwindigkeitskurve, kurz MGK genannt [8]. Der grau eingefärbte Bereich über der MGK stellt also einen unzulässigen Bereich im Zustandsraum dar, welcher nicht durchlaufen werden darf. Da in der Formel (4.13) die Steigung des Pfads im Nenner steht, kann die MGK in manchen Bereichen auch unendlich große Werte einnehmen. Dies ist im Beispiel dort dargestellt, wo die MGK den Zustandsraum nach oben hin verlässt. Die minimale Pfadgeschwindigkeit \dot{s}_{\min} muss nicht aufgetragen werden, da diese zur s -Achse achsensymmetrisch zu \dot{s}_{\max} ist. Im Zustandsraum wird aber nur der Bereich oberhalb der s -Achse betrachtet, da $s(t)$ nur so monoton steigend wird.

Beim Integrationsvorgang ist folgendes Vorgehen üblich [5]. Begonnen wird bei $s = 0$ und $\dot{s} = 0$. Enden muss die Integration bei $s = s_{\text{Ende}}$ und $\dot{s} = 0$. Dies ist notwendig, weil aus (4.2) ein \dot{s} von null auch einer Gelenkwinkelgeschwindigkeit von null entspricht. Dies ist vor allem dann notwendig, wenn die Ableitung des Pfads an dieser Stelle ungleich null ist. Auch hier ist die Forderung von Abschnitt 3.1 gültig, dass die Geschwindigkeit am Anfang und am Ende null sein muss. In dem gezeigten Beispiel ist $s_{\text{Ende}} = 1$.

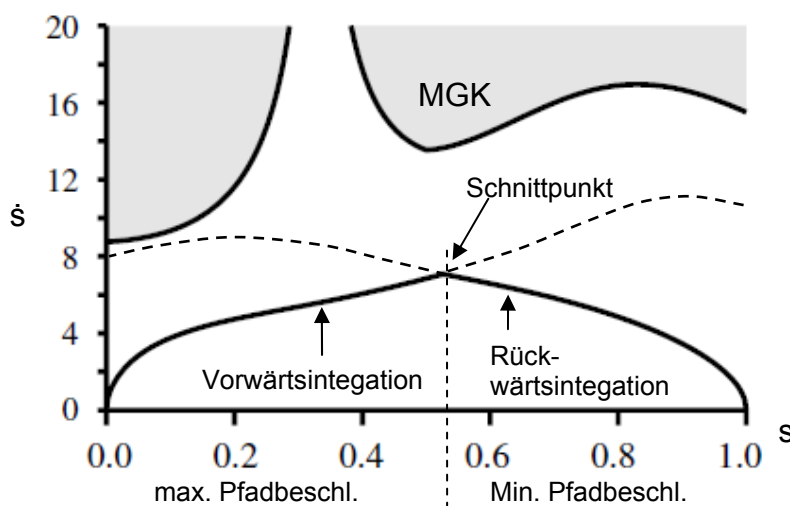


Abbildung 4.9: Vorwärts- und Rückwärtsintegration im Zustandsraum [5]

Es wird vom Ursprung des Koordinatensystems aus mit \dot{s}_{\max} integriert. Einfache Integration ergibt \dot{s} , nochmaliges Integrieren bringt s . Es wird immer mit einem festen Zeitschritt integriert. Für jeden Integrationsschritt wird ein neues \dot{s}_{\max} ermittelt, da dies von s und \dot{s} abhängig ist. Es wird deshalb immer mit \dot{s}_{\max}

integriert um eine möglichst hohe Bahngeschwindigkeit \dot{s} zu erhalten und dadurch die Verfahrdauer auf der Bahn zu verkürzen, also die Trajektorie zeitoptimal zu ermitteln. Dieser Vorgang wird so lange durchgeführt, bis die Maximale-Geschwindigkeitskurve geschnitten wird oder der Zustandsraum bei $s > s_{\text{Ende}}$ verlassen wird. Tritt einer der beiden Fälle ein, so wird von $s = s_{\text{Ende}}$ und $\dot{s} = 0$ aus mit \ddot{s}_{min} rückwärts integriert. Auch dieser Vorgang wird so lange gemacht, bis die MGK geschnitten wird oder das durch Integration ermittelte $s_{\text{neu}} < 0$ wird. Schneiden sich die beiden integrierten Kurven, so muss der Schnittpunkt genau ermittelt werden. Da über die Zeit integriert wurde, lässt sich auch die Zeit für die Vorwärtsintegration sowie für die Rückwärtsintegration bis zum Schnittpunkt ermitteln. Durch Verbindung der beiden Integrationen ergibt sich so ein $s(t)$, mit dem vom Pfad eine Trajektorie errechnet werden kann, die die Grenzen einhält. Die Trajektorie, die Geschwindigkeit und die Beschleunigung des Gelenks kann mit Nutzung der integrierten Kurve direkt mit den Formeln (4.1) bis (4.3) berechnet werden.

Treffen sich die Vorwärts- und die Rückwärtsintegration nicht, sondern wird davor die MGK geschnitten, so bedarf es zusätzlicher Schritte. Ein Übertreten der MGK würde bedeuten, dass die maximale Geschwindigkeit vom Roboter überschritten wird. Von dem Punkt s_{MGK} , an dem die Vorwärtsintegration die MGK schneidet, wird jetzt weiter auf der MGK gelaufen. Es wird jetzt der Punkt s_{Tangente} gesucht, an dem bei Integration mit \ddot{s}_{max} die MGK nicht überschritten, sondern nur tangential berührt wird. Von diesem Punkt aus wird weiter mit \ddot{s}_{max} vorwärts integriert und mit \ddot{s}_{min} rückwärts integriert. Dieses Vorgehen ist in Abbildung 4.10 dargestellt. Schneidet die Rückwärtsintegration vom Punkt s_{Tangente} die erste Vorwärtsintegration von $s = 0$, so ist dieser Schnittpunkt s_{SP1} der Umschaltunkt von der maximalen zur minimalen Bahnbeschleunigung. Schneidet die Vorwärtsintegration von s_{Tangente} aus die Rückwärtsintegration von $s = s_{\text{Ende}}$, so bedeutet auch der Schnittpunkt s_{SP2} ein Umschalten von maximaler zur minimaler Bahnbeschleunigung. Sollte die anfängliche Vorwärts- oder Rückwärtsintegration nicht getroffen werden, so ist dieser Vorgang so oft zu wiederholen, bis ein vollständiger Verlauf von $s = 0$ bis $s = s_{\text{Ende}}$ durch Integration mit minimalen und maximalen \ddot{s} erreicht wird. Eine Trajektorie, die durch

Integration mit Maximalwerten zustande kommt, wird im englischen auch „bang-bang trajectory“ genannt.

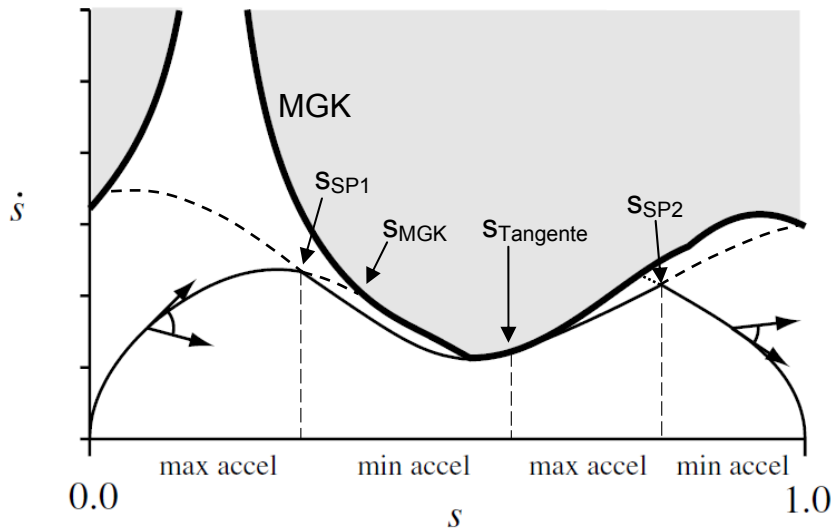


Abbildung 4.10: Ermittlung der Umschaltunkte um Schneiden der MGK zu verhindern [5]

Mit dieser Methode können auch mehrere Achsen berücksichtigt werden. Dazu wird von jeder Achse die MGK berechnet und von all diesen die minimale verwendet. Ebenso wird für jede Achse ein \dot{s}_{\max} berechnet und auch hier der kleinste Wert verwendet. Bei \dot{s}_{\min} muss der größte Wert berücksichtigt werden.

Ein weiterer Vorteil, den dieses Vorgehen bietet, ist, dass das dynamische Modell des Roboters in die Berechnung mit einfließen kann und dadurch berücksichtigt wird. Die Standardform für die Roboterdynamik lautet folgendermaßen.

$$\tau_M = M(q)\ddot{q} + C(q, \dot{q}) + g(q) \quad (4.17)$$

Dabei werden mit der Matrix M die Massenträgheiten berücksichtigt, C ist ein Vektor für die geschwindigkeitsabhängigen Coriolis- und Zentripetalkräfte und g ist ein Vektor mit den Gravitationskräften. Das Zeichen Tau (τ_M) steht für das Drehmoment des Antriebs. Wird diese Gleichung nach $\ddot{q}(q, \dot{q})$ umgestellt und anstelle von \ddot{q}_{\max} bzw. \ddot{q}_{\min} in (4.15) bzw. (4.16) eingesetzt, so kann für die komplette Berechnung das dynamische Modell berücksichtigt werden. Die Grenze \ddot{q}_{\max} bzw. \ddot{q}_{\min} ist nun kein fester Wert mehr, sondern abhängig von der aktuellen Position des Roboters auf dem geplanten Pfad und dem maximalen Drehmoment des Antriebs.

Als großer Nachteil der entkoppelten Trajektorienplanung muss wohl gelten, dass hier nur die maximale Gelenkwinkelbeschleunigung und nicht der Ruck berücksichtigt wird. Es ist zwar denkbar, dass anstatt mit \ddot{s}_{\max} mit $\dot{\ddot{s}}_{\max}$ integriert wird, allerdings muss dann noch \ddot{s}_{\max} berücksichtigt werden. Im Gegensatz zu \dot{s}_{\max} , das in Form der MGK aufgetragen wird, ist \ddot{s}_{\max} von s und von \dot{s} abhängig. Das heißt es wäre zur Veranschaulichung ein 3-D-Plot nötig. Das Problem liegt im Schnittpunkt von Vorwärts- zur Rückwärtsintegration. Auch in diesem Übergang muss der Ruck berücksichtigt werden. Zum aktuellen Zeitpunkt existiert für dieses Problem keinerlei Literatur und eine Lösung des Problems konnte, in der für die Anfertigung der Bachelorarbeit verfügbaren Zeit, nicht gefunden werden.

4.4 Beispiel einer Trajektorie

Werden nun mehrere Achsen geplant so ergibt sich der folgende Verlauf des Pfades.

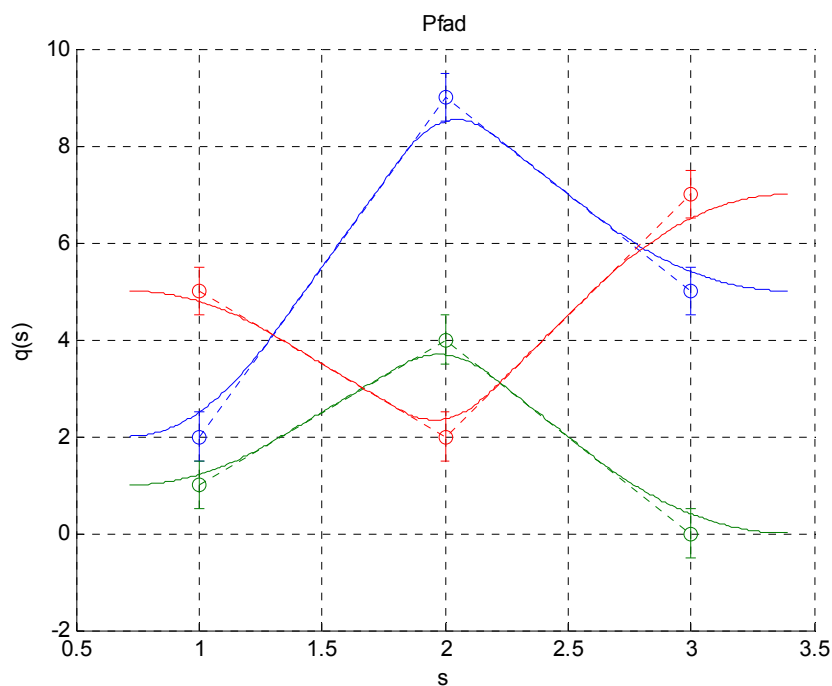


Abbildung 4.11: Geplanter Pfad für mehrere Achsen

Durch den Integrationsvorgang kann nun das anschließend dargestellte $s(t)$ ermittelt werden.

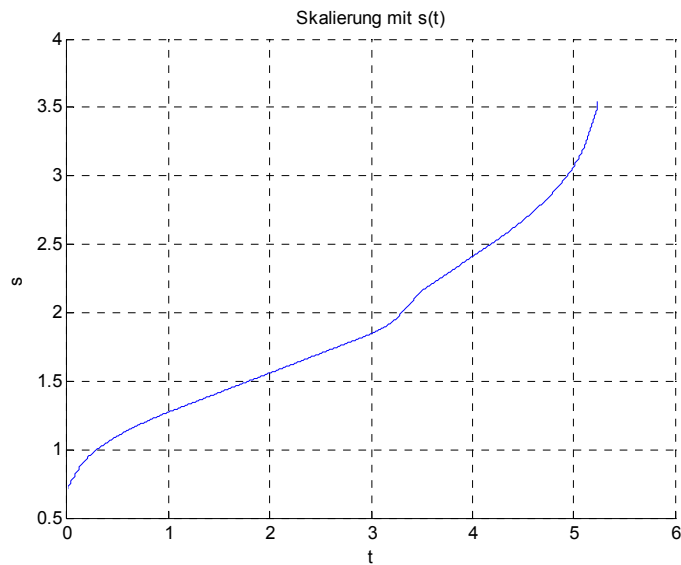


Abbildung 4.12: Skalierung mit $s(t)$

Werden mit diesem $s(t)$ die Trajektorien ermittelt, so ergibt sich der in Abbildung 4.13 gezeigte Verlauf.

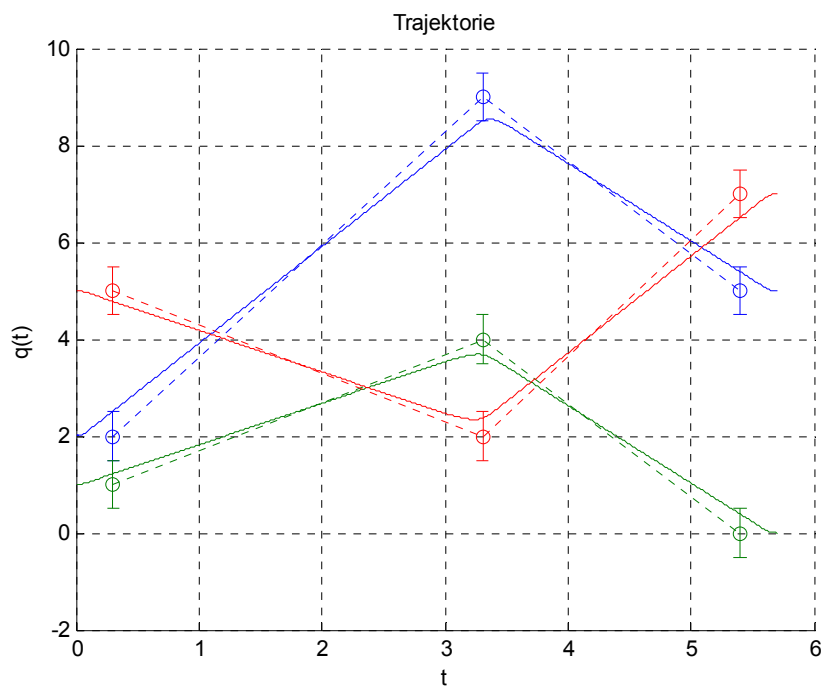


Abbildung 4.13: Neu ermittelte Trajektorien

5 Umsetzung in einen Algorithmus

Zur Entwicklung eines Algorithmus wird das Softwarepaket Matlab verwendet. Dieses Programm wurde speziell zur Lösung mathematischer Probleme und zur grafischen Darstellung der Ergebnisse entwickelt. Es ist im Besonderen für numerische Berechnungen mithilfe von Matrizen ausgelegt. Kleinere Programme können durch die mathematisch orientierte Syntax der Matlab-Skriptsprache wesentlich einfacher als mit der Programmiersprache C erstellt werden. Besonders der Zugriff auf die umfangreichen Funktionsbibliotheken stellt eine große Erleichterung dar. Ein weiterer Vorteil ist die sehr einfache Fehlersuche mit dem Debug-Modus [9].

5.1 Polynom-Generator

Mit dem Polynom-Generator wird der Pfad $q(s)$ für jede Achse erzeugt. Dabei wird bei der Pfad-Erzeugung darauf geachtet, dass diese zueinander voll-synchronisiert sind.

5.1.1 Vorgehen zur Pfaderzeugung

Dieser Programmteil bekommt von der Übergabe des Planers für jede Achse die gleiche Anzahl an Soll-Winkelstellungen. Zu jeder Winkelstellung wird auch die maximal zulässige Toleranz übergeben. Die Toleranz kann unterschiedlich sein, da eine gegebene Toleranz bei den unteren Achsen, also denen, die näher an der Basis des Roboters sind, eine größere Abweichung des Endeffektors zur Folge hat als bei einem Gelenk nahe am Endeffektor. Von den unteren Achsen ist eventuell, je nach Roboterkonfiguration, der gesamte Roboteraufbau abhängig, der mehrere Meter betragen kann. Die Achse direkt vor dem Endeffektor ist unter Umständen nur noch für eine Länge von einigen Zentimetern des Roboterarms zuständig. Die Toleranz bei den Via-Punkten kann auch variieren, da bei manchen Bewegungen mehr Präzision erforderlich ist als bei anderen.

Der Generator trägt also jeden Winkel q über ganze natürliche Werte n über die x -Achse auf, welche dem Pfadparameter s entspricht. Außerdem wird das Toleranzband berechnet. Dazu werden abschnittsweise definierte Geradengleichungen aufgestellt.

Die allgemeine Form lautet:

$$y = m x + t \quad (5.1)$$

Jede Gleichung ist nur für den Bereich

$$[n; n+1] \quad \text{mit } n \in N \quad (5.2)$$

gültig. Anstatt x wird hier abschnittsweise über s_{Gerade} gerechnet. Für die Gleichungen muss gelten:

$$s_{\text{Gerade}} = [0; 1] \quad (5.3)$$

Für die ganz normale Verbindung zwischen zwei Via-Punkten gilt:

$$y_n = \frac{q_{n+1} - q_n}{s_{n+1} - s_n} s_{\text{Gerade}} + q_n = \frac{q_{n+1} - q_n}{1} s_{\text{Gerade}} + q_n = (q_{n+1} - q_n) s_{\text{Gerade}} + q_n \quad (5.4)$$

Die Gleichung für die obere Grenze des Toleranzbandes wird durch Addition der Toleranz an den Via-Punkten ermittelt.

$$y_{n+} = ((q_{n+1} + \text{Toleranz}_{n+1}) - (q_n + \text{Toleranz}_n)) s_{\text{Gerade}} + (q_n + \text{Toleranz}_n) \quad (5.5)$$

Durch Subtraktion wird die Gleichung für die untere Grenze des Toleranzbandes ermittelt.

$$y_{n-} = ((q_{n+1} - \text{Toleranz}_{n+1}) - (q_n - \text{Toleranz}_n)) s_{\text{Gerade}} + (q_n - \text{Toleranz}_n) \quad (5.6)$$

Als Beispiel werden für den Gelenkwinkel q vier Werte vorgegeben. Diese sind 20° , 70° , 70° und 40° . Diese vier Werte werden über den Parameter s aufgetragen. Da s ganze natürliche Zahlen einnimmt, sind das für dieses Beispiel eins, zwei, drei und vier. Dargestellt ist dies in der nachfolgenden Abbildung. Die roten Punkte stellen die Gelenkwinkel dar. Als Besonderheit beträgt hier die Tole-

ranz an den beiden Punkten bei 70° einmal $\pm 2^\circ$, an dem zweiten Punkt $\pm 3^\circ$. An den anderen beiden Punkten (20° , 40°) beträgt die Toleranz $\pm 1^\circ$. Die drei eingezeichneten Linien sind die obere und die untere Grenze des Toleranzbandes sowie die direkte Verbindung zweier Gelenkwinkel. Berechnet werden diese Geraden mit den Formeln (5.5), (5.6) und (5.4). Aufgrund der unterschiedlichen Toleranzen wird jetzt besonders zwischen $s = 2$ und $s = 3$ deutlich, weshalb für die Steigung der oberen und unteren Toleranz nicht einfach die Steigung zwischen zwei Via-Punkten verwendet werden darf.

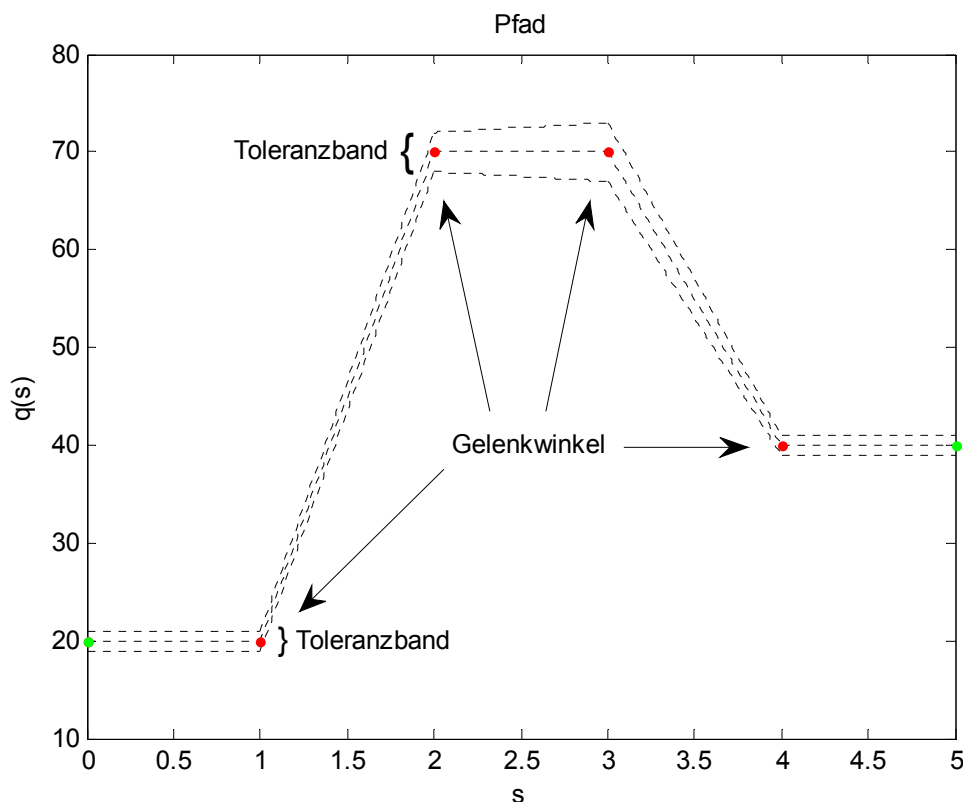


Abbildung 5.1: Gelenkwinkel mit Toleranzband

Wie außerdem in dem gezeigten Beispiel zu erkennen ist, ist der erste und der letzte Punkt dupliziert worden. In der Abbildung sind diese beiden Punkte grün eingezeichnet. Dies ist nötig, um am Anfang und am Ende des Verfahrens eine Steigung von null zu erhalten. Eine Steigung von null bedeutet auch eine Gelenkwinkelgeschwindigkeit von null. Dadurch wird die Bedingung erfüllt, dass die Geschwindigkeit des Gelenks am Anfang und am Ende null ist, es sich also aus dem Stillstand heraus in Rotation versetzt und am Ende wieder zum Stehen

kommt. Durch Vorgabe einer anderen Anfangs- oder Endsteigung könnte so beispielsweise ein Fahrweg geplant werden, der bereits eine bestimmte Startgeschwindigkeit besitzt oder dessen Bewegung nicht mit dem Stillstand endet.

Im nächsten Schritt der Pfadplanung werden nun die roten Punkte mit Geraden verbunden. Dies entspricht der einfachsten Verbindungsform und kann abschnittsweise mit der Gleichung (5.4) realisiert werden. Dadurch erhält man bereits die Grundform des Pfades, welcher im Anschluss abgebildet ist.

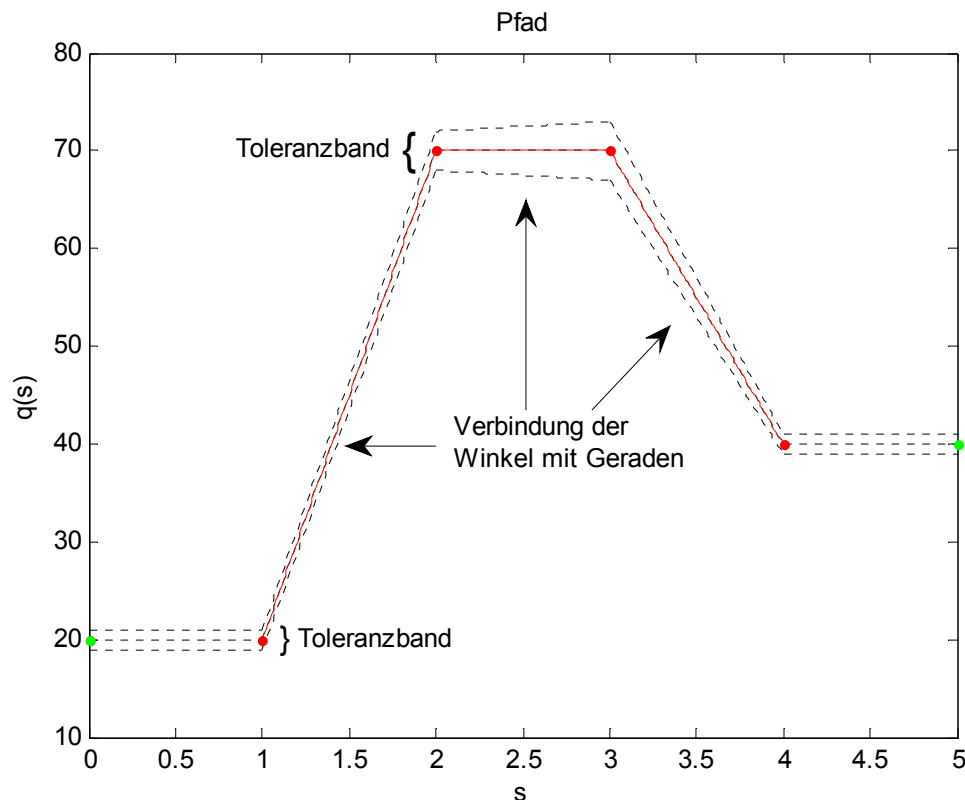


Abbildung 5.2: Verbindung der Gelenkwinkel mit Geraden

In diesem Bild ist bereits eine grundlegende Verbindung der Gelenkwinkel erstellt. Allerdings muss wie in Abschnitt 4.2 beschrieben der Pfad mindestens dreimal ableitbar sein. Dabei müssen die erste und die zweite Ableitung stetig sein. Das heißt, die Ecken der Verbindung an den Gelenkwinkeln müssen mit Bogenstücken abgerundet werden. Dazu muss jedes Bogenstück mindestens aus Polynomen dritten Grades bestehen, um zuvor genannte Bedingung erfüllen zu können.

Beispielhaft werden die Schritte für einen Bogen beim Übergang mit $s = 2$ durchgegangen. Der Beginn des Übergangs ist auf der roten Geraden genau zwischen $s = 1$ und $s = 2$, also bei $s = 1,5$. Das Ende liegt bei $s = 2,5$.

Allgemein gilt für den Beginn und das Ende eines Bogens am Via-Punkt n folgender Zusammenhang:

Beginn des Bogenstücks ist beim Schnittpunkt A (SP A)

$$s_{SP A} = n - 0,5 \quad (5.7)$$

Das Ende ist bei Schnittpunkt B (SP B)

$$s_{SP B} = n + 0,5 \quad (5.8)$$

Das Bogenstück wird mit der Funktion $f(s_{\text{Bogen}})$ bezeichnet, die Gerade mit $g(s_{\text{Gerade}})$. Das Bogenstück wird nur für diesen Abschnitt definiert und kann Werte von null bis eins für s_{Bogen} einnehmen. Der erste Übergang von der Geraden auf das Bogenstück ist in der Abbildung 5.3 mit Schnittpunkt A gekennzeichnet. Dort müssen folgende Bedingungen erfüllt werden.

$$\begin{aligned} f_n(0) &= g_n(0,5) \\ f'_n(0) &= g'_n(0,5) \\ f''_n(0) &= g''_n(0,5) = 0 \end{aligned} \quad (5.9)$$

Für den zweiten Übergang bei Schnittpunkt B gilt dann:

$$\begin{aligned} f_n(1) &= g_{n+1}(0,5) \\ f'_n(1) &= g'_{n+1}(0,5) \\ f''_n(1) &= g''_{n+1}(0,5) = 0 \end{aligned} \quad (5.10)$$

Diese sechs Bedingungen werden von einem Polynom fünften Grades erfüllt. Diese Funktion besitzt sechs Koeffizienten. Damit kann ein lineares Gleichungssystem mit sechs Gleichungen und sechs Unbekannten, den gesuchten Koeffizienten, aufgestellt werden. Ein solches Gleichungssystem kann mit Matlab sehr einfach mit dem „Backslash“-Operator (\backslash) gelöst werden. Wird damit ein Kurvenstück berechnet und eingezeichnet, ergibt sich der in rot dargestellte Verlauf. Dieses Teilstück verlässt allerdings das Toleranzband.

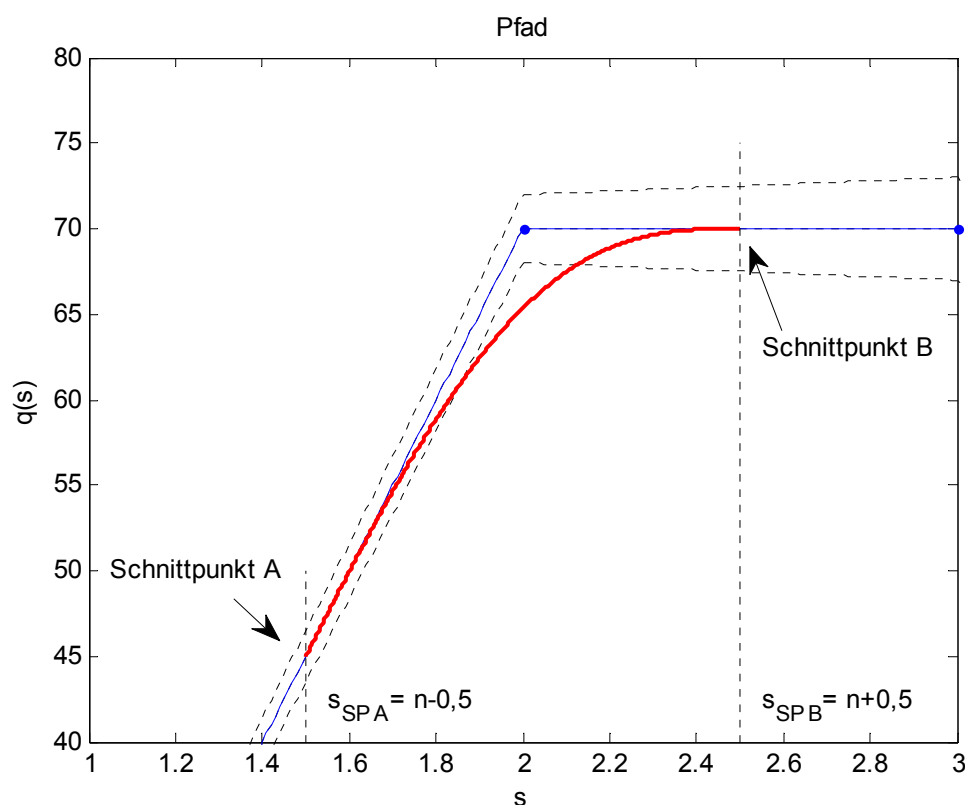


Abbildung 5.3: Verbindung von zwei Geraden mit Kurvenstück

Ein Einhalten der Toleranz kann durch iteratives Einrücken des Bogenstücks erreicht werden. Dazu wird das Bogenstück auf jeder Seite um einen festen Wert eingerückt, im Beispiel in der folgenden Abbildung um $s = 0,1$. Das Bogenstück hat damit nicht mehr die Länge eins. Der Beginn von $f_n(0)$ ist jetzt nicht mehr bei $s = 1,5$ sondern bei $s = 1,5 + \Delta s$, oder allgemein bei

$$s = n - 0,5 + \Delta s \quad (5.11)$$

Die Werte der beiden Ableitungen von (5.9) und (5.10) bleiben gleich, lediglich der Anfangswert von $f_n(0)$ und der Endwert von $f_n(s_{\text{Bogen}})$ ändern sich.

$$f_n(0) = g_n(0,5 + \Delta s) \quad (5.12)$$

Dabei ist der Endwert von $f_n(s_{\text{Bogen}})$ bei $s_{\text{Bogen}} = 1 - 2\Delta s$.

$$f_n(1 - 2\Delta s) = g_{n+1}(0,5 - \Delta s) \quad (5.13)$$

Die zweite Gerade am Schnittpunkt B beginnt jetzt bei

$$s = n + 0,5 - \Delta s \quad (5.14)$$

Das Einrücken wird so lange ausgeführt, bis die Toleranz eingehalten wird. Insgesamt wird der Bogen auf jeder Seite um $\Delta s = 0,3$ verkürzt. Das iterative Verkürzen und das daraus resultierende Bogenstück sind unten abgebildet.

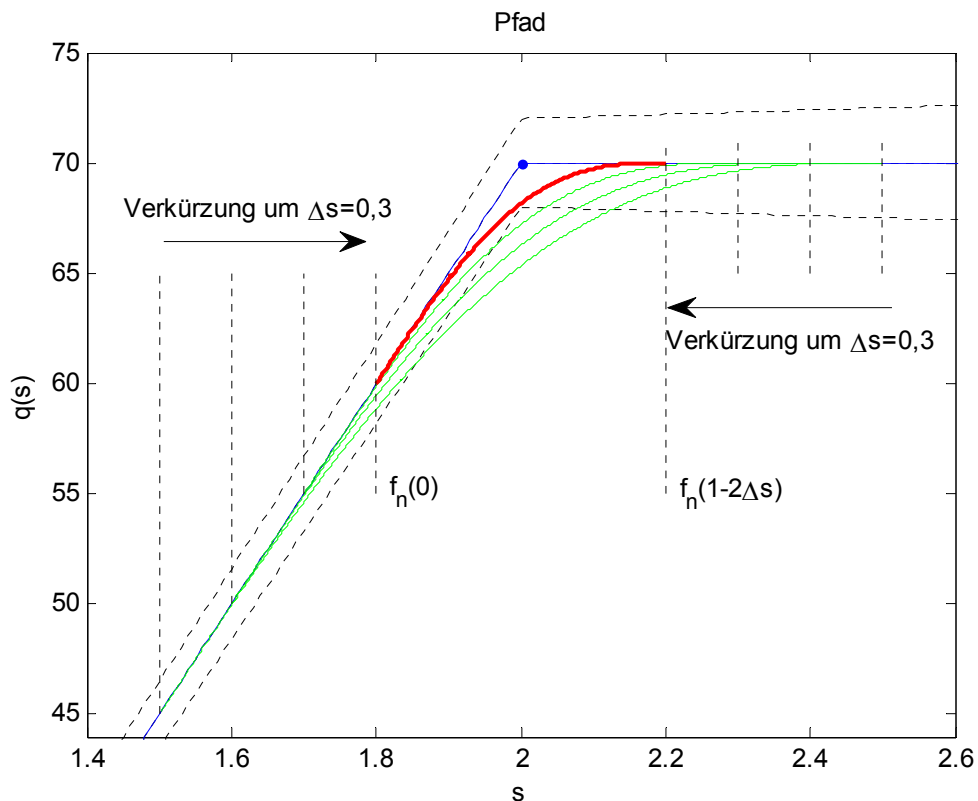


Abbildung 5.4: Iteratives Verkürzen des Kurvenstücks bis Toleranzband eingehalten wird

Mit diesem Vorgehen kann für jede Verbindung von zwei Geraden ein Kurvenstück gefunden werden. Der Pfad kann nun abschnittsweise aus Geraden und Polynomen fünften Grades gebildet werden.

5.1.2 Bogenstück mit zwei Polynomen dritten Grades

Das soeben beschriebene Vorgehen ist eine Möglichkeit für die Pfadplanung. Da sich ein iteratives Vorgehen allerdings negativ auf die Rechenzeit auswirkt und allgemein ein Umgang mit Funktionen niedrigerer Ordnung leichter fällt, wird nun ein anderes Verfahren vorgestellt. Das Kurvenstück wird nun nicht mehr aus einer, sondern aus zwei Funktionen zusammengesetzt. In dem hier

vorgestellten Fall bestehen die zwei Funktionen aus Polynomen dritten Grades. Jedes Bogenstück ist für das gleiche Intervall von s links und rechts vom Via-Punkt gültig. Dabei beginnt das erste Bogenstück bei maximal $s = n-0,5$ und endet bei $s = n$. Das Zweite ist maximal im Bereich von $s = n$ bis $s = n+0,5$ gültig. Wenn diese Bogenstücke die Toleranz nicht einhalten, muss bei diesem Ansatz nicht iterativ nach innen gerückt werden, sondern der Beginn und damit auch das Ende kann analytisch gelöst werden. Die zwei Teilbogenstücke sind unten dargestellt, die Berechnung der beiden folgt im Anschluss.

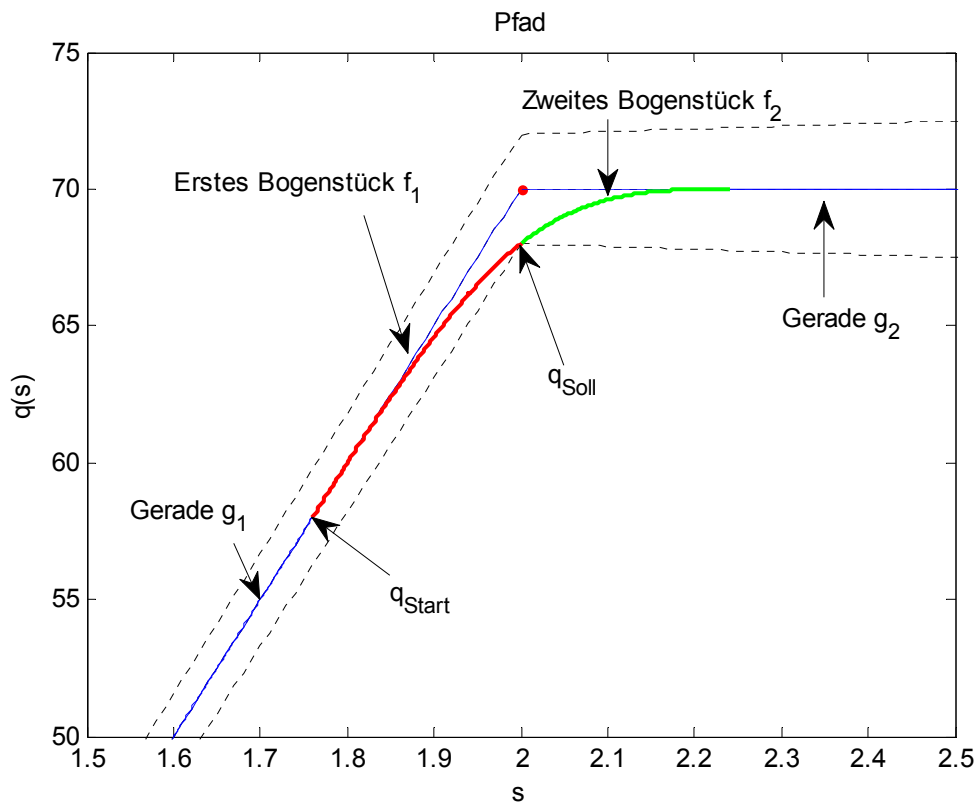


Abbildung 5.5: Kurvenstück aus zwei Polynomen dritten Grades

Das erste Bogenstück wird mit f_1 , das zweite mit f_2 bezeichnet. Die Funktion f_1 wird über den Parameter s_{f1} berechnet, f_2 über s_{f2} .

$$f_1(s_{f1}) = a_{13}s_{f1}^3 + a_{12}s_{f1}^2 + a_{11}s_{f1} + a_{10} \quad (5.15)$$

$$f_2(s_{f2}) = a_{23}s_{f2}^3 + a_{22}s_{f2}^2 + a_{21}s_{f2} + a_{20} \quad (5.16)$$

Diese beiden Gleichungen besitzen acht Koeffizienten. Damit können insgesamt acht Bedingungen erfüllt werden. Die ersten drei Bedingungen muss f_1 für

$s_{f1} = 0$ erfüllen. Das s_f kann für beide Funktionen Werte von $s_f = 0$ bis $s_{f \text{ Ende}}$, maximal ist $s_{f \text{ Ende}} = 0,5$ einnehmen. Da die Geradengleichungen auch über den Parameter s berechnet werden, werden diese ab jetzt mit s_g bezeichnet. Jede Gerade ist für ein Intervall von null bis eins für s_g gültig.

$$g_{1/2}(s_g) = m_{1/2} s_g + t_{1/2} \quad (5.17)$$

Der Übergang von g_1 auf f_1 ist für die Gerade bei $s_g = 1 - s_{f1 \text{ Ende}}$. Eingezeichnet ist er in Abbildung 5.5 bei q_{Start} . Der Übergang von f_2 auf g_2 ist bei $s_g = s_{f2 \text{ Ende}}$.

$$\begin{aligned} f_1(0) &= g_1(s_g) \\ f_1'(0) &= g_1'(s_g) \\ f_1''(0) &= g_1''(s_g) = 0 \end{aligned} \quad (5.18)$$

Im Übergang zwischen den einzelnen Teilbogenstücken muss gelten:

$$\begin{aligned} f_1(s_{f1 \text{ Ende}}) &= f_2(0) = q_{\text{Soll}} \\ f_1'(s_{f1 \text{ Ende}}) &= f_2'(0) \\ f_1''(s_{f1 \text{ Ende}}) &= f_2''(0) \end{aligned} \quad (5.19)$$

Für den Übergang vom zweiten Teilbogenstück auf die zweite Gerade g_2 muss gelten:

$$\begin{aligned} f_2'(s_{f2 \text{ Ende}}) &= g_2'(s_{f2 \text{ Ende}}) \\ f_2''(s_{f2 \text{ Ende}}) &= g_2''(s_{f2 \text{ Ende}}) = 0 \end{aligned} \quad (5.20)$$

Die Bedingung, dass

$$f_2(s_{f2 \text{ Ende}}) = g_2(s_{f2 \text{ Ende}}) \quad (5.21)$$

wird automatisch erfüllt, die Beweisführung folgt später in diesem Abschnitt.

Die Berechnung von $s_{f \text{ Ende}}$ berücksichtigt das Einrücken der Bogenstücke. Außerdem wird dadurch das Einhalten der Toleranz berücksichtigt. Die Berechnung erfolgt folgendermaßen.

Die beiden Ableitungen von f_1 lauten:

$$f_1'(s_{f1}) = 3a_{13}s_{f1}^2 + 2a_{12}s_{f1} + a_{11} \quad (5.22)$$

$$f_1''(s_{f1}) = 6a_{13}s_{f1} + 2a_{12} \quad (5.23)$$

Für f_2 lauten sie:

$$f_2'(s_{f2}) = 3a_{23}s_{f2}^2 + 2a_{22}s_{f2} + a_{21} \quad (5.24)$$

$$f_2''(s_{f2}) = 6a_{23}s_{f2} + 2a_{22} \quad (5.25)$$

Da beide Funktionen über dasselbe Intervall von s gültig sind, gilt für beide als Endwert s_{Ende} .

Auflösen von (5.18) ergibt:

$$a_{10} = m_1(1 - s_{\text{Ende}}) + t_1 \quad (5.26)$$

$$a_{11} = m_1 \quad (5.27)$$

$$a_{12} = 0 \quad (5.28)$$

Mit (5.20) ergibt sich:

$$3a_{23}s_{\text{Ende}}^2 + 2a_{22}s_{\text{Ende}} + a_{21} = m_2 \quad (5.29)$$

$$6a_{23}s_{\text{Ende}} + 2a_{22} = 0 \quad \Leftrightarrow \quad a_{23} = \frac{-a_{22}}{3s_{\text{Ende}}} \quad (5.30)$$

Der Sollwert, den die Bogenstücke im Übergang einnehmen sollen, ist der Via-Punkt mit der maximal möglichen Toleranz. Eingezeichnet ist dieser Wert bei q_{Soll} . Die Formeln von (5.19) ergeben ausgeschrieben:

$$a_{13}s_{\text{Ende}}^3 + a_{12}s_{\text{Ende}}^2 + a_{11}s_{\text{Ende}} + a_{10} = q_{\text{Soll}} \quad (5.31)$$

$$3a_{13}s_{\text{Ende}}^2 + 2a_{12}s_{\text{Ende}} + a_{11} = a_{21} \quad (5.32)$$

$$6a_{13}s_{\text{Ende}} + 2a_{12} = 2a_{22} \quad \Leftrightarrow \quad a_{22} = 3a_{13}s_{\text{Ende}} + a_{12} \quad (5.33)$$

Wird (5.30), (5.32) und (5.33) in (5.29) eingefügt, so ergibt sich:

$$a_{13} = \frac{m_2 - a_{11} - 3a_{12}s_{\text{Ende}}}{6s_{\text{Ende}}^2} \quad (5.34)$$

(5.27), (5.28) und (5.34) in (5.31) ergibt:

$$\frac{m_2 + 5m_1}{6} s_{\text{Ende}} + a_{10} = q_{\text{Soll}} \quad (5.35)$$

Wird nun a_{10} mit (5.26) ersetzt, so kann nun s_{Ende} berechnet werden.

$$s_{\text{Ende}} = 6 \frac{q_{\text{Soll}} - m_1 - t_1}{m_2 - m_1} \quad (5.36)$$

Dabei ist t_1 der Wert des vorherigen Via-Punktes.

Mit dieser Formel kann also das Einrücken der Bogenstücke analytisch berechnet werden. Zu beachten ist dabei, dass s_{Ende} nie größer als 0,5 werden darf.

Als nächstes folgt die Beweisführung von (5.21). Für die Koeffizienten des ersten Polynoms gelten die Bedingungen (5.26), (5.27), (5.28) und (5.34). Für das zweite Polynom gilt für a_{21} (5.32) und, da a_{20} gleich q_{Soll} ist, kann dafür (5.31) verwendet werden. Für a_{22} und a_{23} müssen die Gleichungen von (5.20) umgestellt werden. Die zweite Ableitung ergibt a_{22} :

$$6a_{23}s_{\text{Ende}} + 2a_{22} = 0 \Leftrightarrow a_{22} = -3a_{23}s_{\text{Ende}} \quad (5.37)$$

Die erste Ableitung ergibt ausgeschrieben:

$$3a_{23}s_{\text{Ende}}^2 + 2a_{22}s_{\text{Ende}} + a_{21} = m_2 \quad (5.38)$$

(5.37) in diesen Ausdruck eingesetzt und nach a_{23} aufgelöst ergibt:

$$a_{23} = \frac{m_2 - a_{21}}{-3s_{\text{Ende}}^2} \quad (5.39)$$

Wird nun in diese Gleichung a_{21} eingesetzt, kann a_{23} und damit auch a_{22} ermittelt werden. Werden nun diese vier Koeffizienten in die Gleichung (5.16) des zweiten Bogenstücks eingesetzt, so ergibt sich für $f_2(s_{\text{Ende}})$:

$$f_2(s_{\text{Ende}}) = m_2 s_{\text{Ende}} + m_1 + t_1 \quad (5.40)$$

Der Startwert der Geraden g_2 ist der Endwert von g_1 , also $g_1(1)$. Dadurch gilt:

$$\begin{aligned} t_2 &= g_1(1) \\ t_2 &= m_1 \times 1 + t_1 \end{aligned} \quad (5.41)$$

Damit kann die Beweisführung abgeschlossen werden:

$$\begin{aligned}
 f_2(s_{\text{Ende}}) &= g_2(s_{\text{Ende}}) \\
 m_2 s_{\text{Ende}} + m_1 + t_1 &= m_2 s_{\text{Ende}} + t_2 \\
 m_2 s_{\text{Ende}} + t_2 &= m_2 s_{\text{Ende}} + t_2
 \end{aligned}
 \tag{5.42}$$

Die Pfadplanung mit diesem Vorgehen ist im aktuellen Programm implementiert.

5.1.3 Erzeugung voll-synchroner Achsen

Als nächstes muss noch volle Synchronität der Achsen sichergestellt werden. Dazu muss jede Achse genau über die gleichen Intervalle von s dieselbe Bewegungsform ausführen. In der momentan implementierten Pfadplanung sind das die gleichmäßige Bewegung und die beschleunigte Bewegung. In der nächsten Abbildung ist die Problematik zu erkennen.

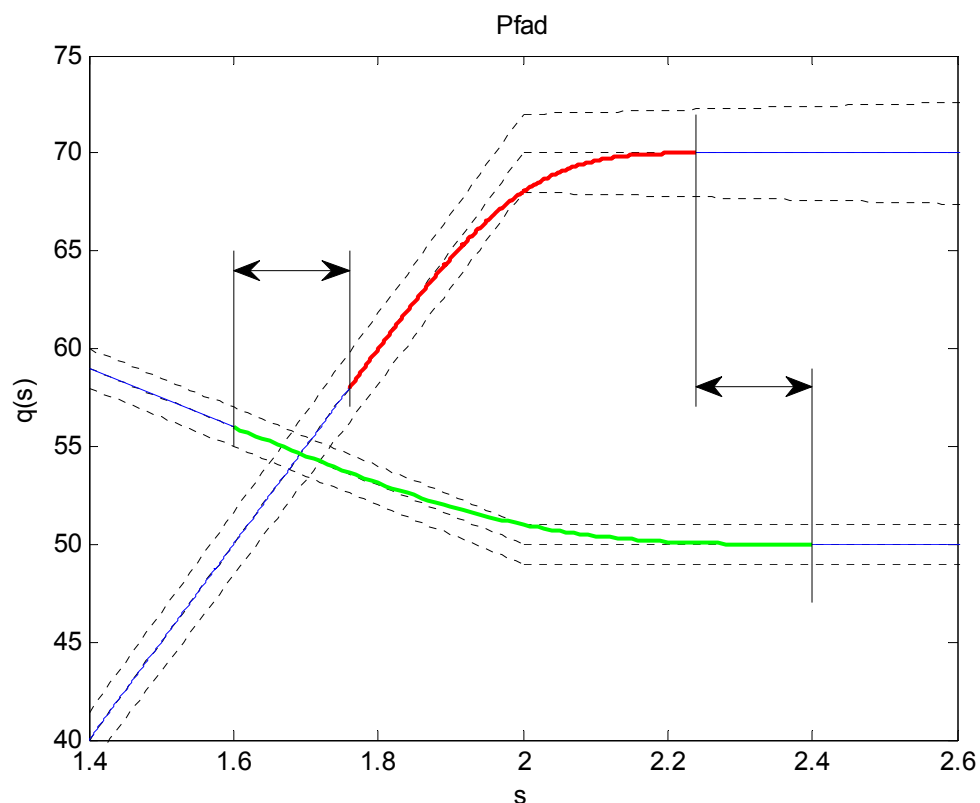


Abbildung 5.6: Nicht synchronisierte Bewegungsphasen bei zwei Achsen

In dem Bildausschnitt ist im Bereich von $s = 1,4$ bis $s = 1,6$ die Bewegungsform bei beiden Achsen eine Gerade, also eine gleichförmige Bewegung. Dann aber

beginnt bei der zweiten Achse die Änderung der Gelenkwinkelgeschwindigkeit, dargestellt mit dem grünen Bogenstück. Die erste Achse, in der Abbildung rot eingezeichnet, ändert die Geschwindigkeit erst bei $s = 1,76$. In den mit den Pfeilen gekennzeichneten Bereichen treten bei den beiden Achsen unterschiedliche Bewegungsphasen auf. Da ein Verlängern des roten Bogens eine Verletzung des Toleranzbandes zur Folge hätte, muss der grüne Bogen durch Verringerung von s_{Ende} verkürzt werden. Dasselbe Vorgehen lässt sich auf beliebig viele Achsen anwenden, wichtig ist dabei nur, dass alle Achsen an das kürzeste Bogenintervall s angepasst werden.

Für die Berechnung der Achsen, die verkürzt werden, gilt folgende Vorgehensweise. Von der kürzesten Achse ist s_{Ende} bekannt. Dies wird für alle restlichen Achsen verwendet. In der Formel (5.19) ändert sich die erste Zeile zu:

$$f_1(s_{\text{Ende}}) = f_2(0) \quad (5.43)$$

Die Koeffizienten a_{10} , a_{11} und a_{12} können wie gewohnt mit den Formeln (5.26), (5.27) und (5.28) berechnet werden. Mit (5.34) wird a_{13} berechnet. Für das zweite Bogenstück berechnen sich die Koeffizienten von a_{21} und a_{22} mit (5.32) und (5.33), für a_{23} gilt (5.30). Der Koeffizient a_{20} kann durch (5.43) wie folgt berechnet werden.

$$a_{13}s_{\text{Ende}}^3 + a_{12}s_{\text{Ende}}^2 + a_{11}s_{\text{Ende}} + a_{10} = a_{20} \quad (5.44)$$

Das Ergebnis für zwei Achsen mit voll-synchronen Bewegungsphasen ist in Abbildung 5.7 dargestellt. Durch dieses Vorgehen wird allerdings bei der grünen Achse das Toleranzband nicht mehr voll ausgenutzt. Das kann zu einer Erhöhung der Pfadbeschleunigung führen, was wiederum bei der Ermittlung der Trajektorien eventuell zu einer Verlängerung der Verfahrdauer des Roboters führen kann.

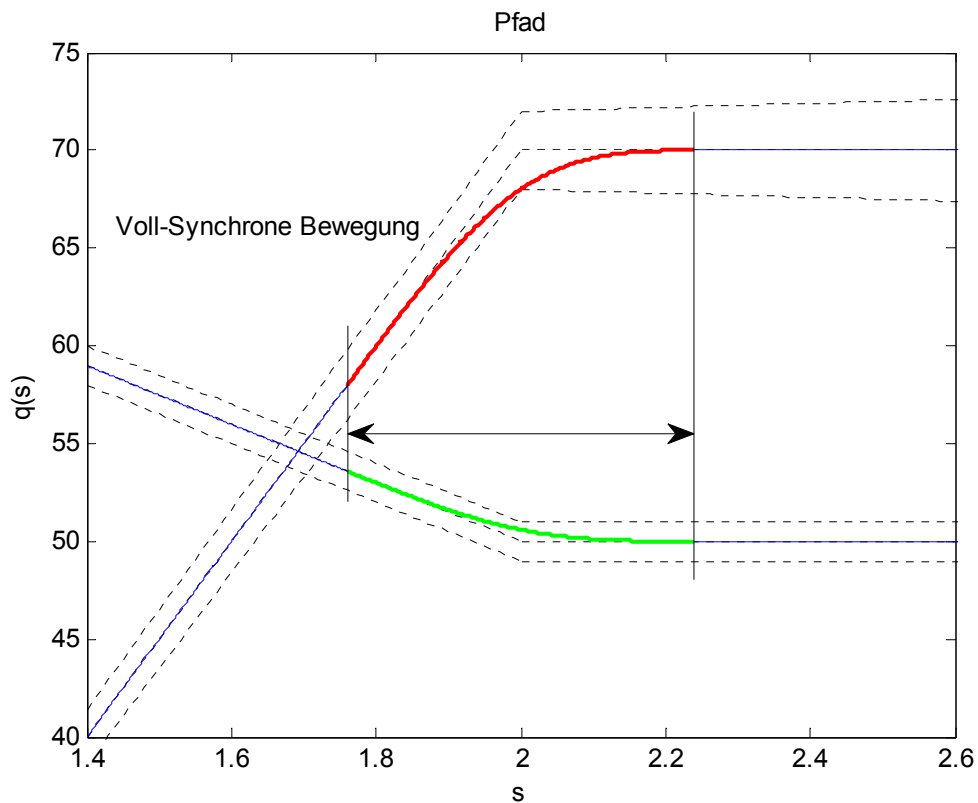


Abbildung 5.7: Voll-Synchrone Bewegungsphasen von zwei Achsen

Durch dieses Vorgehen kann ein Pfad erzeugt werden, der das Toleranzband nicht verlässt und sicherstellt, dass alle Achsen voll-synchrone Bewegungen ausführen.

5.1.4 Optimierung der Krümmung im Übergang

Eine letzte Verbesserung kann den Pfad aber noch weiter optimieren. Werden alle Achsen so wie vorher beschrieben geplant, so kann es vorkommen, dass zwei oder mehrere Bogenstücke direkt ineinander übergehen, also keine gleichförmige Bewegung mehr dazwischen liegt. Ein solcher Fall ist in der folgenden Abbildung dargestellt. Gezeigt wird hier nur eine Achse. Diese Achse durchfährt drei Via-Punkte. Eingezeichnet sind drei Bogenstücke, wobei ein jedes aus zwei Polynomen dritten Grades besteht. Die dazugehörigen Krümmungen sind in der anschließenden Abbildung gezeigt.

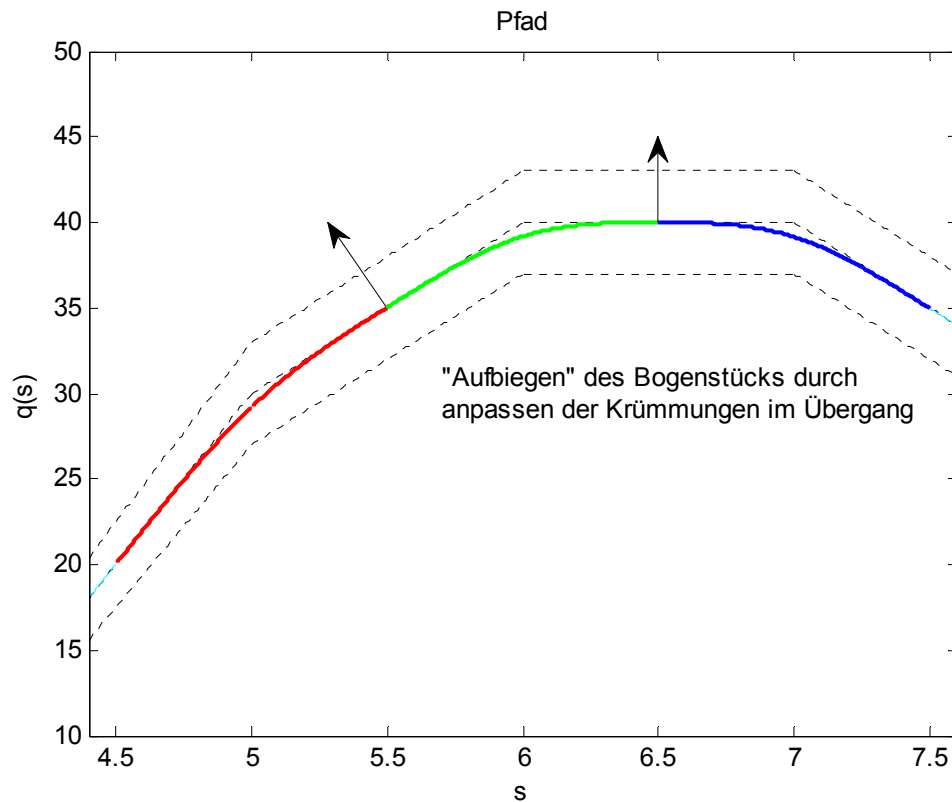


Abbildung 5.8: Optimierung der Krümmung im Übergang zwischen zwei Bogenstücken

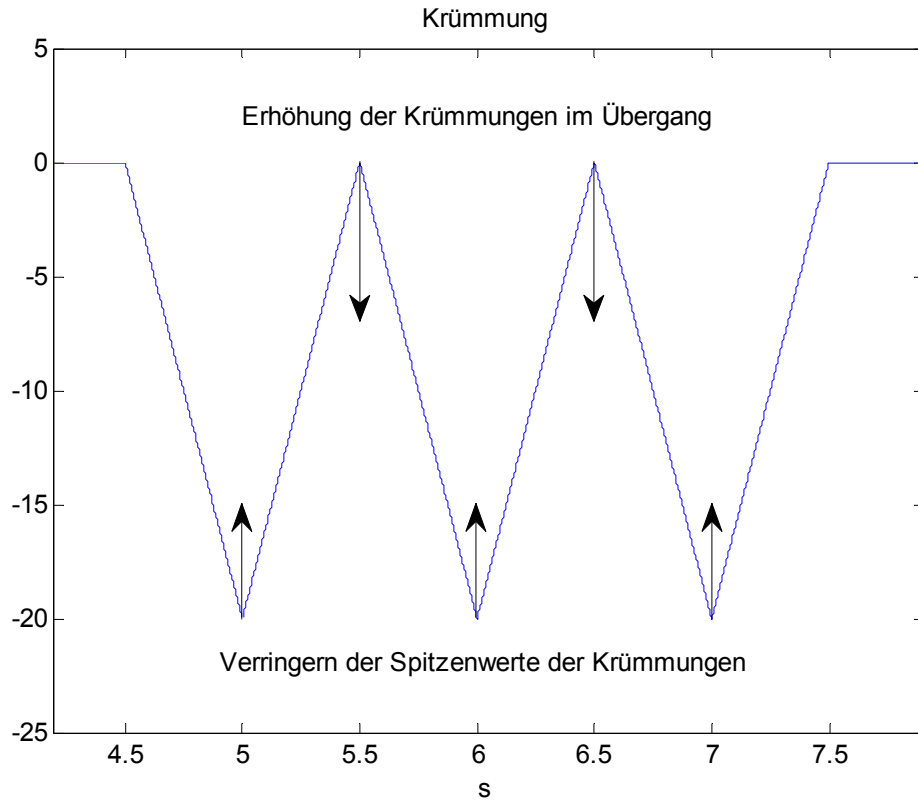


Abbildung 5.9: Verringerung der maximalen Krümmungen durch Anpassung im Übergang

Wegen der zweiten Bedingung von (5.20) ist in Abbildung 5.9 zu erkennen, dass ein jedes Bogenstück am Ende für die Krümmung den Wert null einnehmen muss. Gerade aber bei diesem dargestellten Fall ist dies ungünstig. Besser wäre es die Krümmung im Übergang so zu wählen, dass die Betrags-Maximalwerte geringer werden. Grundlegend für die Krümmungswerte im Übergang ist, dass sie dasselbe Vorzeichen wie die Maximalwerte besitzen müssen. Werden dann die Werte betragsmäßig erhöht, wird eine Verringerung der Spitzenwerte erreicht. Das Vorgehen dieses Prinzips ist in obiger Abbildung zu sehen. Das hat allerdings zur Folge, dass die Bogenstücke „nach außen hin“ gebogen werden, was in Abbildung 5.8 schematisch gezeigt wird.

Zunächst muss der Algorithmus nach den Stellen suchen, an denen $s_{f\text{ Ende}} = 0,5$ ist. Dies kann zum Beispiel mit dem Matlab-Befehl „find“ bewerkstelligt werden. Dadurch ergibt sich eine Liste mit allen Bogenstücken, die ineinander übergehen können. Wichtig sind aber nur die Bogenstücke, die direkt nebeneinander liegen und nicht durch eine Gerade getrennt sind. Es müssen also mindestens zwei Bogenstücke mit diesen Bedingungen direkt nebeneinander liegen. Im nächsten Schritt muss geprüft werden, ob die Krümmungen in diesen aneinander liegenden Bogenstücken dasselbe Vorzeichen besitzen. Dadurch wird sichergestellt, dass sich beide Bögen auch in die gleiche Richtung krümmen. Sind nun die relevanten Bogenstücke erkannt, kann zwischen Übergängen die Krümmung so vorgegeben werden, dass die maximale Krümmung minimiert wird.

Als Nachteile dieser Methode gilt, dass für die Optimierung alle betroffenen Bogenstücke noch einmal neu berechnet werden müssen. Bei diesem Vorgang kann auch das Toleranzband nicht voll ausgenutzt werden. Bei dem Übergang, in dem die Krümmung optimiert wird, ist von den beiden angrenzenden Punkten derjenige mit der geringeren Toleranz maßgebend. Die Berechnung ist nach Abbildung 5.10 und Abbildung 5.11 aufgeführt.

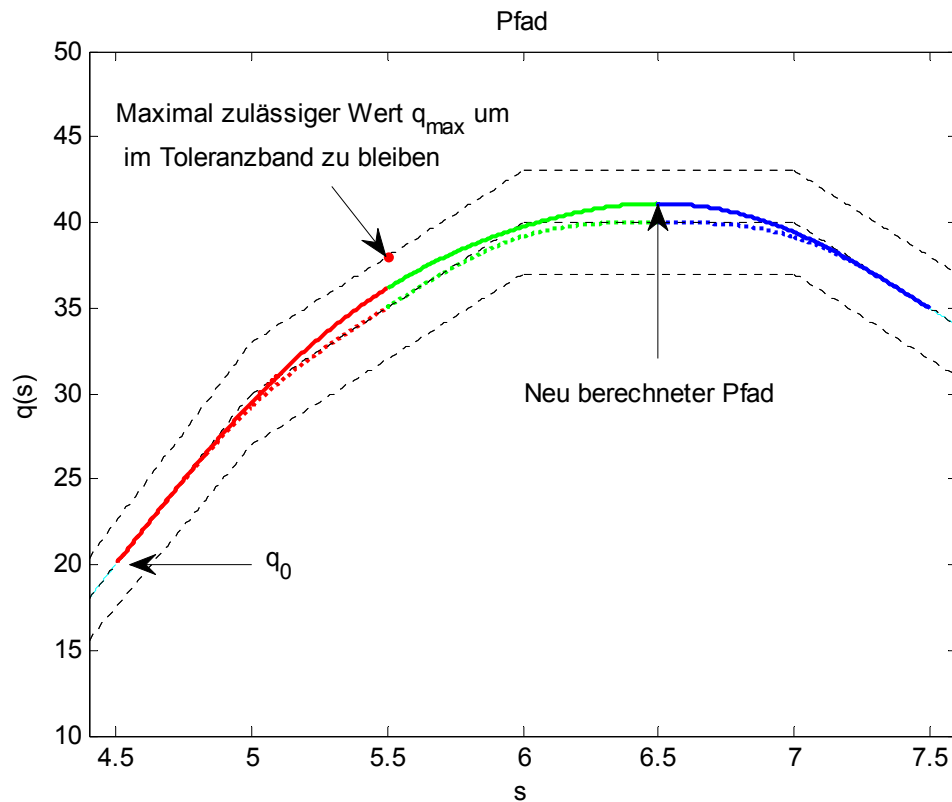


Abbildung 5.10: Neuer Pfad nach Optimierung der Krümmung

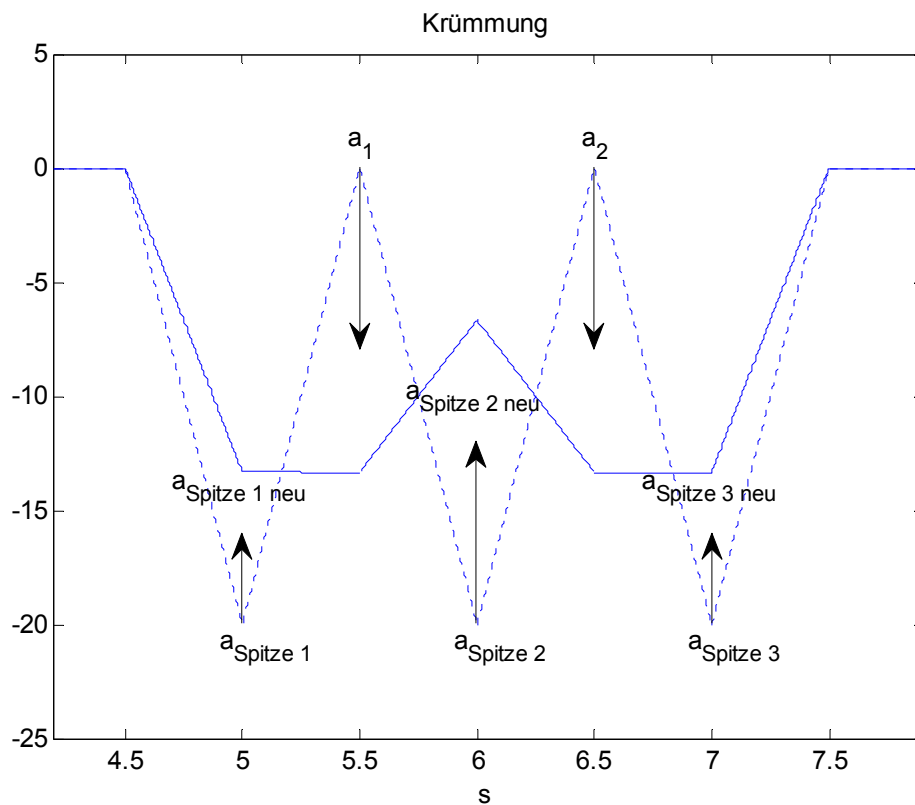


Abbildung 5.11: Krümmungen nach dem Optimierungsvorgang

Um ein Verlassen des Toleranzbandes zu verhindern, muss zunächst die maximal mögliche Krümmung a_{\max} im Übergang berechnet werden, mit der der Pfad innerhalb der gegebenen Grenzen bleibt. In der Abbildung 5.10 ist der maximale Wert q_{\max} markiert. Die Steigung der Geraden g_1 , von der aus das Bogenstück startet, wird mit m_1 bezeichnet. Die nächste Gerade, die die zwei Via-Punkte miteinander verbindet, heißt g_2 , die dazugehörige Steigung m_2 . Außerdem wichtig ist der Startwert des ersten Bogenstückes, bezeichnet mit q_0 . Es gilt $s = 0,5$. Die Berechnung von a_{\max} lautet:

$$a_{\max} = \frac{q_0 + m_1 s + m_2 s - q_{\max}}{s^2} \quad (5.45)$$

In Abbildung 5.11 wird die Krümmung im Übergang 1 mit a_1 bezeichnet, im Übergang 2 mit a_2 . Diese beiden Krümmungen müssen also kleiner als a_{\max} sein, da sonst das Toleranzband übertreten wird. Die Spitzenwerte werden a_{Spitze1} , a_{Spitze2} und a_{Spitze3} genannt. Der Wert, den die Krümmung im Übergang einnimmt, wird zu gleichen Teilen von den beiden Spitzenwerten links und rechts davon abgezogen. Die zwei äußeren Spitzen verringern sich also um:

$$\begin{aligned} a_{\text{Spitze 1 neu}} &= a_{\text{Spitze 1}} - 0,5 a_1 \\ a_{\text{Spitze 3 neu}} &= a_{\text{Spitze 3}} - 0,5 a_2 \end{aligned} \quad (5.46)$$

Auf die Spitze in der Mitte hat sowohl a_1 als auch a_2 Einfluss. Der neue Spitzenwert berechnet sich deshalb mit:

$$a_{\text{Spitze 2 neu}} = a_{\text{Spitze 2}} - 0,5 a_1 - 0,5 a_2 \quad (5.47)$$

Dabei sind zwei Dinge zu beachten. Zum einen darf die Krümmung des ganzen betrachteten Kurvenstücks das Vorzeichen nicht ändern. Dies würde geschehen wenn a_1 oder a_2 so groß wird, dass einer oder beide nebenstehenden Spitzenwerte so weit verringert werden, dass sie das Vorzeichen ändern. Darum muss gelten:

$$a_1 \leq 2 a_{\text{Spitze 1}} \quad \wedge \quad a_1 \leq 2 a_{\text{Spitze 2}} \quad (5.48)$$

Außerdem darf der Wert, der für a_1 und a_2 festgelegt wird, nicht größer sein als der benachbarte, neu berechnete Spitzenwert. Wie auch Abbildung 5.11 zeigt,

wird die Krümmung im Übergang nur so weit erhöht, dass dieser Wert mit dem neuen Spitzenwert genau gleich groß ist. Also:

$$a_1 \leq a_{\text{Spitze 1 neu}} \quad (5.49)$$

Aus Gleichung (5.46) folgt:

$$a_1 \leq a_{\text{Spitze 1}} - 0,5 a_1 \Leftrightarrow a_1 \leq \frac{2}{3} a_{\text{Spitze 1}} \quad (5.50)$$

Das gleiche gilt natürlich auch für den Spitzenwert $a_{\text{Spitze 2}}$ rechts von a_1 .

Werden diese Bedingungen beachtet, so ergibt sich der in Abbildung 5.10 dargestellte, neu optimierte Verlauf des Pfads. Die eben vorgestellten Gleichungen können auch auf Bogenstücke mit mehreren zusammenhängenden Teilen angewandt werden. Im vorliegenden Algorithmus wird dabei folgende Vorgehensweise angewandt. Zunächst werden so wie oben beschrieben alle zusammenhängenden und relevanten Bogenstücke gesucht. Anschließend wird der größte Spitzenwert der Krümmung gesucht und dessen zwei benachbarten Krümmungen im Übergang optimiert. Anschließend wird das nächste Krümmungsmaximum gesucht und nun dessen benachbarten Übergangskrümmungen optimiert.

Für die Berechnung eines Bogenstücks werden wieder die bekannten zwei Polynome dritten Grades verwendet. Allerdings wird den 2. Ableitungen von (5.18) und (5.20) nicht der Wert null, sondern ein neuer Wert a_1 und a_2 zugewiesen. Dadurch ändern sich drei Koeffizienten der beiden Polynome. Diese lauten nun:

$$a_{12} = a_1 \quad (5.51)$$

$$a_{13} = \frac{m_2 - m_1 - \frac{3}{2} a_1 s - \frac{1}{2} a_2 s}{s^2} \quad (5.52)$$

$$a_{23} = \frac{a_2 - a_{22}}{s} \quad (5.53)$$

Mit $s = 0,5$.

Ein Beispiel für einen erstellten Pfad ist in der anschließenden Abbildung gezeigt. Dabei ist beim roten Pfad, im letzten Bogenstück zwischen $s = 5$ und $s = 6$, der Verlauf mit optimierter Krümmung zu erkennen. Jeder Pfad beginnt

mit dem Bogenstück am Punkt $s = 1$. Dadurch startet der Pfad allerdings nicht bei $s = 0$, sondern bei $s = 1 - s_{\text{Ende}}$, wobei s_{Ende} abhängig vom Bogenstück ist und mit (5.36) berechnet wird.

Für das nächste Teilprogramm „Integration“ werden die Werte für den Pfad in Vektoren gespeichert und übergeben. Die Werte, die übergeben werden, weichen in der Skalierung über den Pfadparameter s insofern ab, dass sie bei $s = 0$ beginnen. Das ist in Abbildung 5.13 links dargestellt, rechts sind außerdem die Ableitungen zu sehen, deren Werte auch an den nächsten Programmteil übergeben werden.

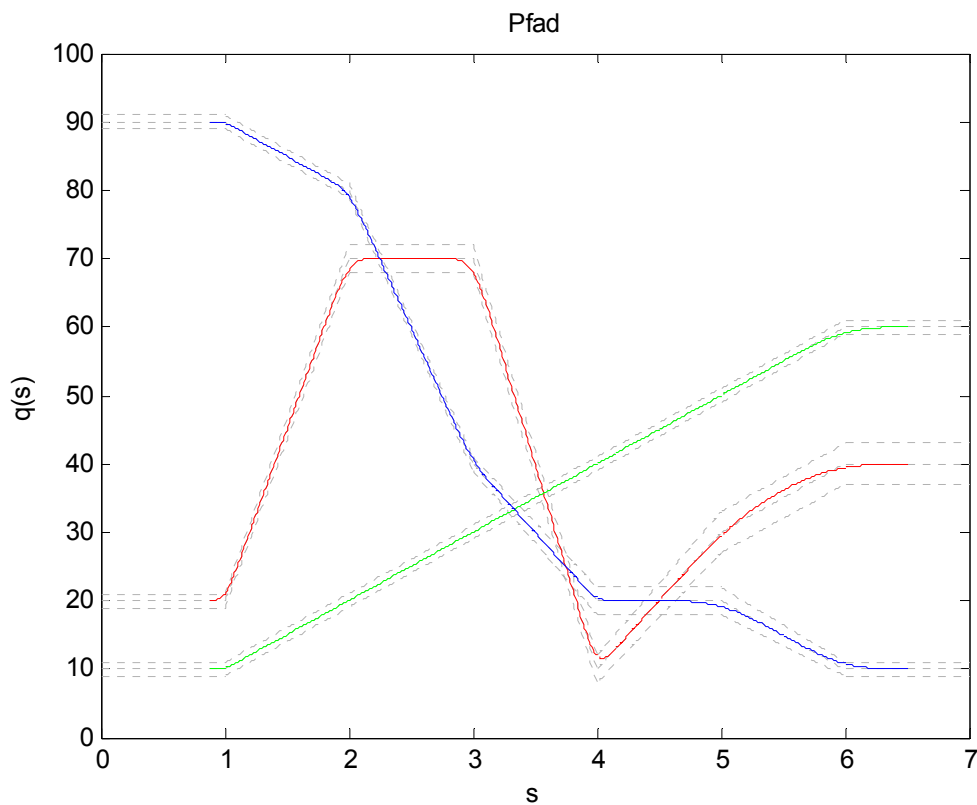


Abbildung 5.12: Beispiel eines geplanten Pfads von drei Achsen

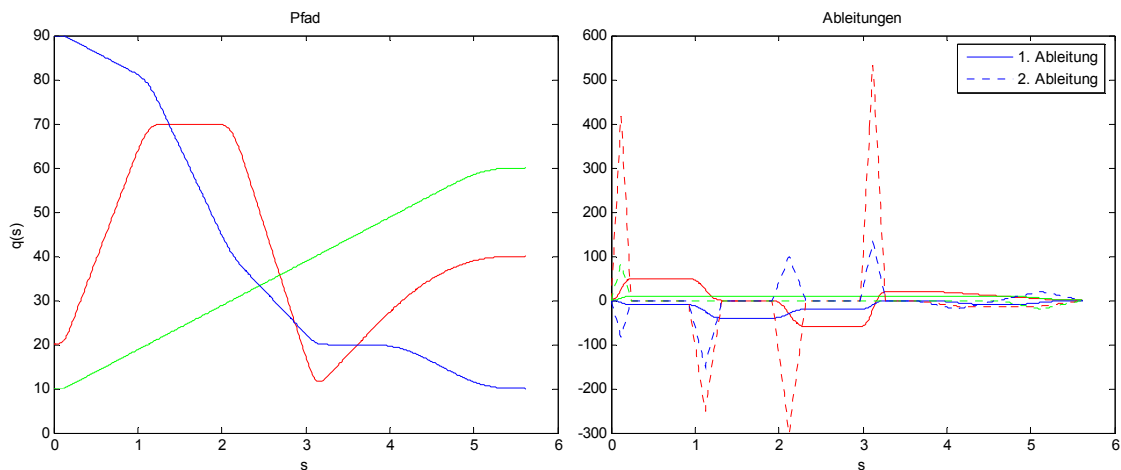


Abbildung 5.13: Pfad und Ableitungen neu skaliert, so dass Start bei $s = 0$

5.2 Integration

In diesem Programmteil wird die Skalierung durch Integration ausgeführt. Benötigt werden dazu die Werte des Pfads $q(s)$ einer jeden Achse und die jeweils dazugehörige erste und zweite Ableitung. Ein Beispiel für den Pfad von zwei Achsen mit den Ableitungen ist nachfolgend dargestellt.

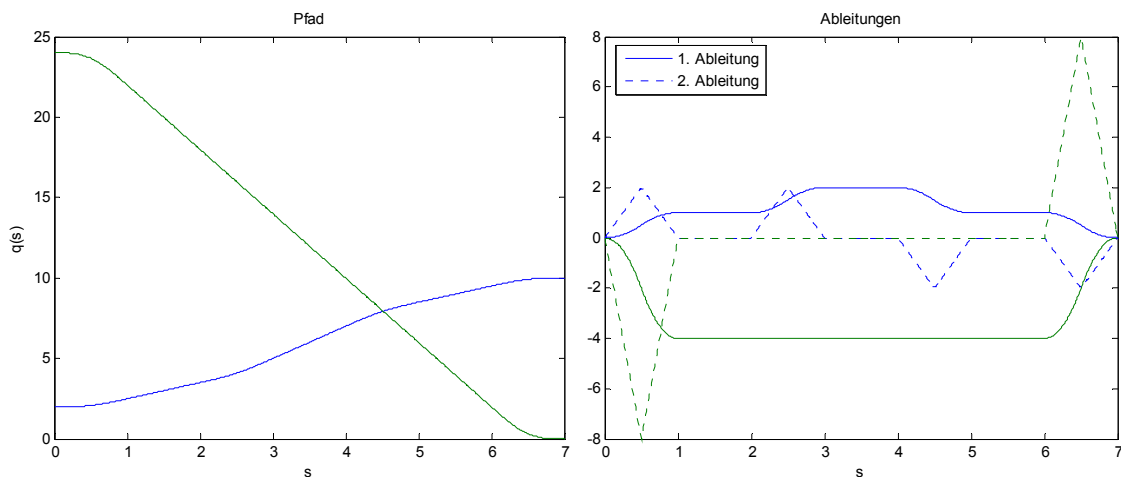


Abbildung 5.14: Beispiel von Pfad und Ableitungen für Integrationsvorgang

Für die weiteren Berechnungen sind Werte für die maximale Gelenkwinkelgeschwindigkeit und –beschleunigung notwendig. Für die Vorstellung des weiteren Algorithmus wird für $\dot{q}_{\max} = 10^\circ/\text{s}$, für $\dot{q}_{\min} = -10^\circ/\text{s}$ angenommen. Für \ddot{q}_{\max} werden $20^\circ/\text{s}^2$ und für \ddot{q}_{\min} $-20^\circ/\text{s}^2$ angenommen. Die Maximale-Geschwindigkeitskurve kann mit (4.13) berechnet werden. In Abbildung 5.15 sind die Kurven

für beide Achsen eingezeichnet. Da aber nur die kleinste MGK benötigt wird, ist diese rot gestrichelt dargestellt. Diese liegt komplett auf der MGK der grünen Achse, da diese am meisten eingebremst werden muss. Für die nächsten Schritte wird nur noch die resultierende MGK verwendet.

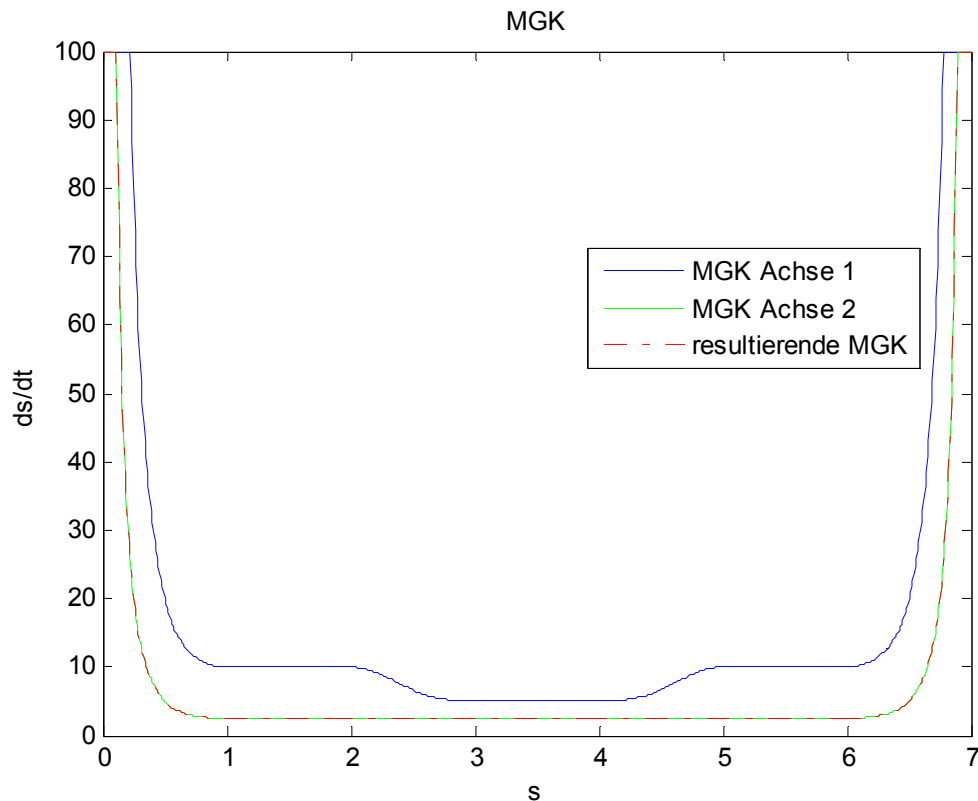


Abbildung 5.15: Maximale-Geschwindigkeitskurve im Zustandsraum

In diesem Diagramm wird nun auch die Vorwärts- und die Rückwärtsintegration durchgeführt. Dazu wird numerisch integriert. Für die Vorwärtsintegration gilt:

$$\dot{s}_{n+1} = \ddot{s}_n t + \dot{s}_n \quad (5.54)$$

Außerdem gilt:

$$s_{n+1} = 0,5 \ddot{s}_n t^2 + \dot{s}_n t + s_n \quad (5.55)$$

Um zu prüfen, ob sich der neue Schritt s_{n+1} noch unter der MGK befindet, muss beachtet werden, dass s in dem gleichen Schritt nach vorne integriert wird, wie auch die Werte von der MGK vorhanden sind. In dem vorgestellten Fall ist für die MGK für jedes $\Delta s = 0,01$ ein Wert vorhanden. Die Integrationszeit muss also

so gewählt werden, dass s auch immer eine Schrittweite von 0,01 einhält. Dazu wird aus (5.55) die Zeit t mit der Diskriminante berechnet.

$$t_{1/2} = \frac{-\dot{s}_n \pm \sqrt{\dot{s}_n^2 - 4 \times 0,5\ddot{s}_n(s_n - s_{n+1})}}{2 \times 0,5\ddot{s}_n} \quad (5.56)$$

Von diesen zwei Zeiten wird die positive, oder sofern beide positiv sind, die kleinere genommen.

Das größte Problem bei der Berechnung der MGK und beim Integrieren stellt die Division durch Null dar. Dies tritt immer dann auf, wenn die Steigung des Pfads null ist. Die MGK kann, wie in der Abbildung 5.16 bei null und bei sieben gezeigt, limitiert werden. Allerdings muss auch \ddot{s} begrenzt werden. Gültige Grenzen, die jeden Fall abhandeln, konnten in der Entstehungszeit der Arbeit noch nicht gefunden werden.

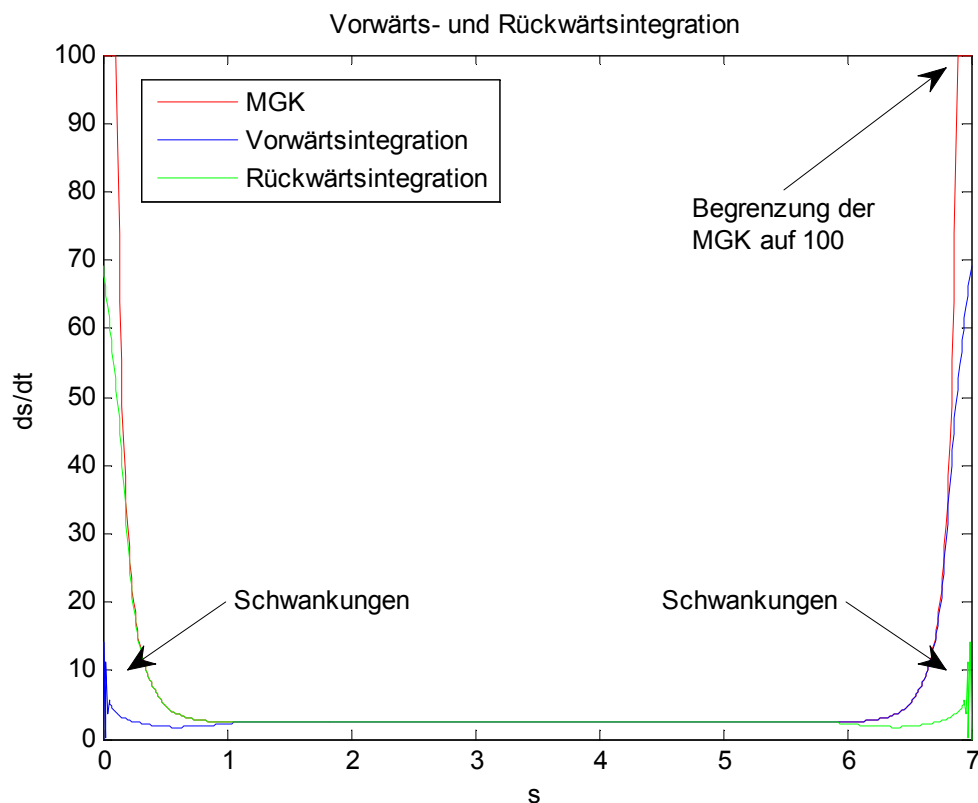


Abbildung 5.16: Vorwärts- und Rückwärtsintegration im Zustandsraum

In der obigen Abbildung ist die MGK mit der Vorwärts- und der Rückwärtsintegration zu sehen. Hier sind bei $s = 0$ und $s = 7$ starke Schwankungen beim

Integrieren zu erkennen. Diese beruhen auf der nicht optimalen Limitierung von \ddot{s} . Als nächstes müssen die beiden Integrationen miteinander kombiniert werden, damit ein geschlossener Verlauf von $s = 0$ bis $s = 7$ entsteht. In diesem Fall wird von $s = 0$ bis $s = 5,87$ die Vorwärtsintegration verwendet. Erst wenn die Rückwärtsintegration kleinere Werte als die Vorwärtsintegration liefert, wird diese benützt. In diesem Beispiel also von $s = 5,87$ bis $s = 7$.

Nun können die einzelnen Integrationszeiten der Vorwärtsintegration mit denen der Rückwärtsintegration addiert werden und es ergibt sich der gezeigte Verlauf von $s(t)$.

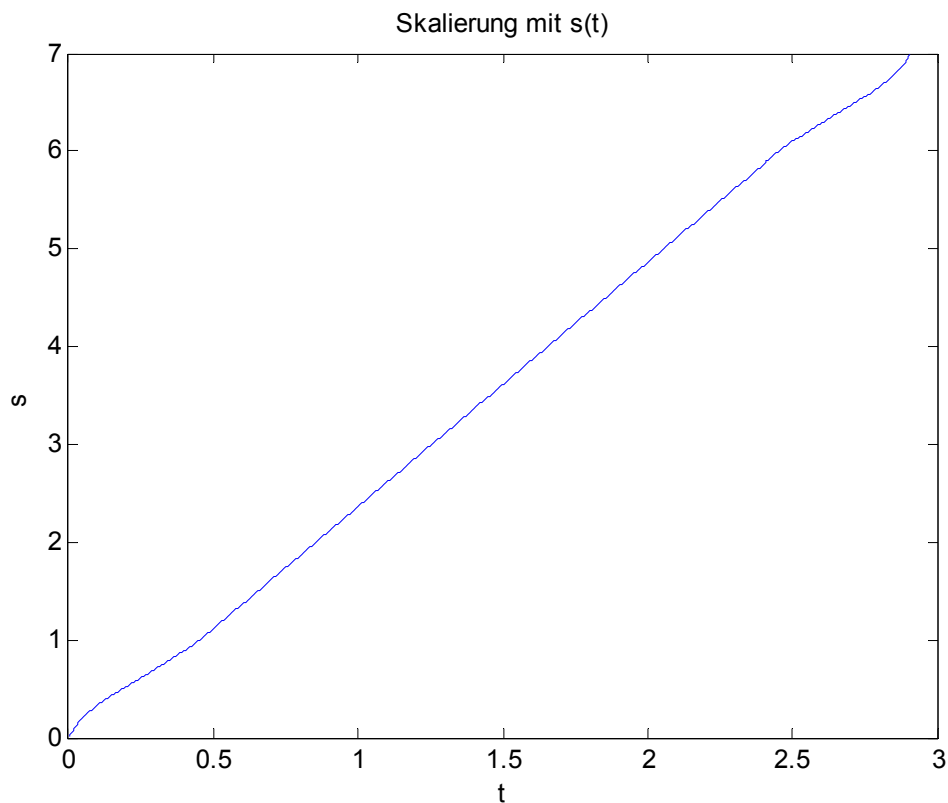


Abbildung 5.17: Skalierung mit $s(t)$

Mit diesem $s(t)$ können nun die Trajektorien und die dazugehörigen Ableitungen ermittelt werden. Diese sind im Anschluss gezeigt.

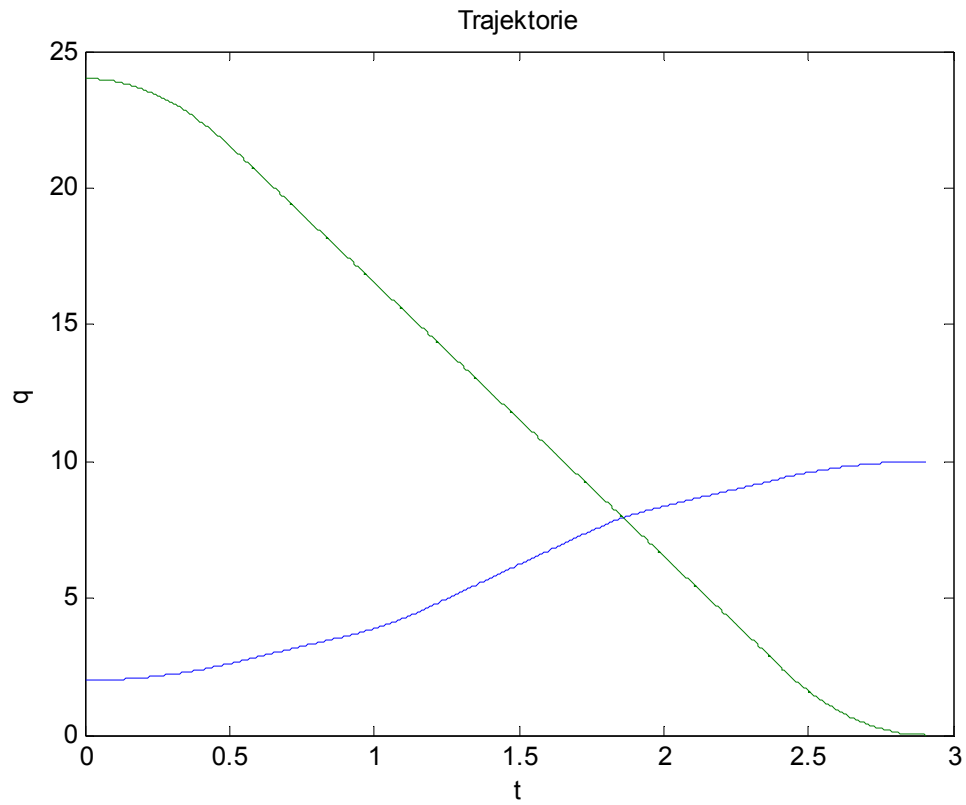


Abbildung 5.18: Resultierende Trajektorien nach der Skalierung

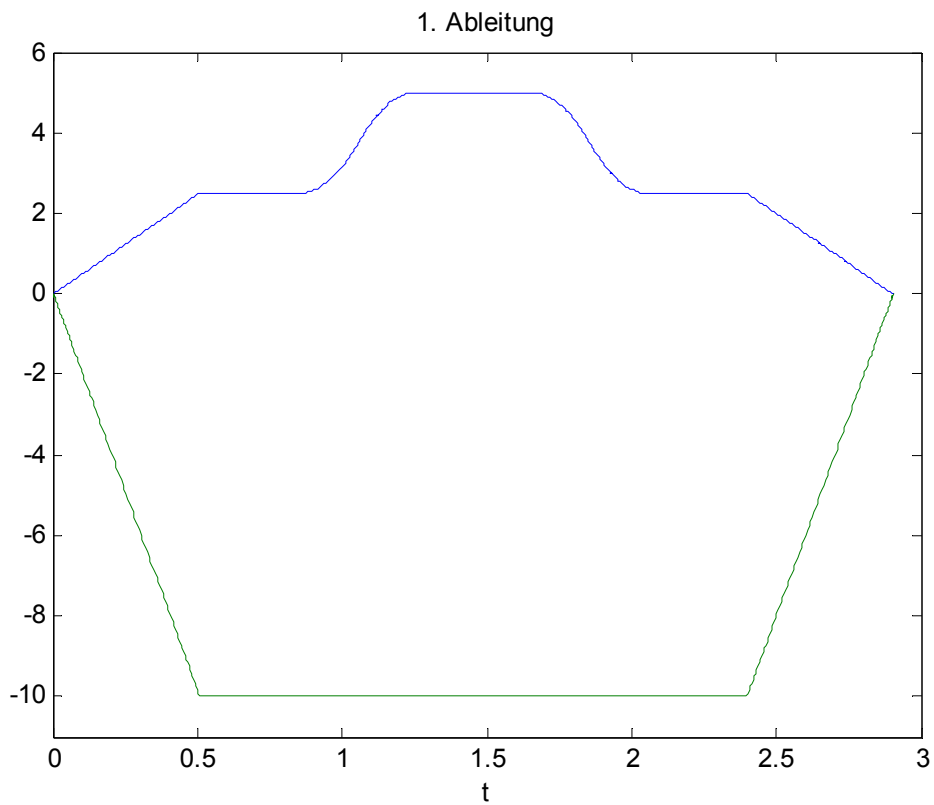


Abbildung 5.19: Gelenkwinkelgeschwindigkeiten nach der Skalierung

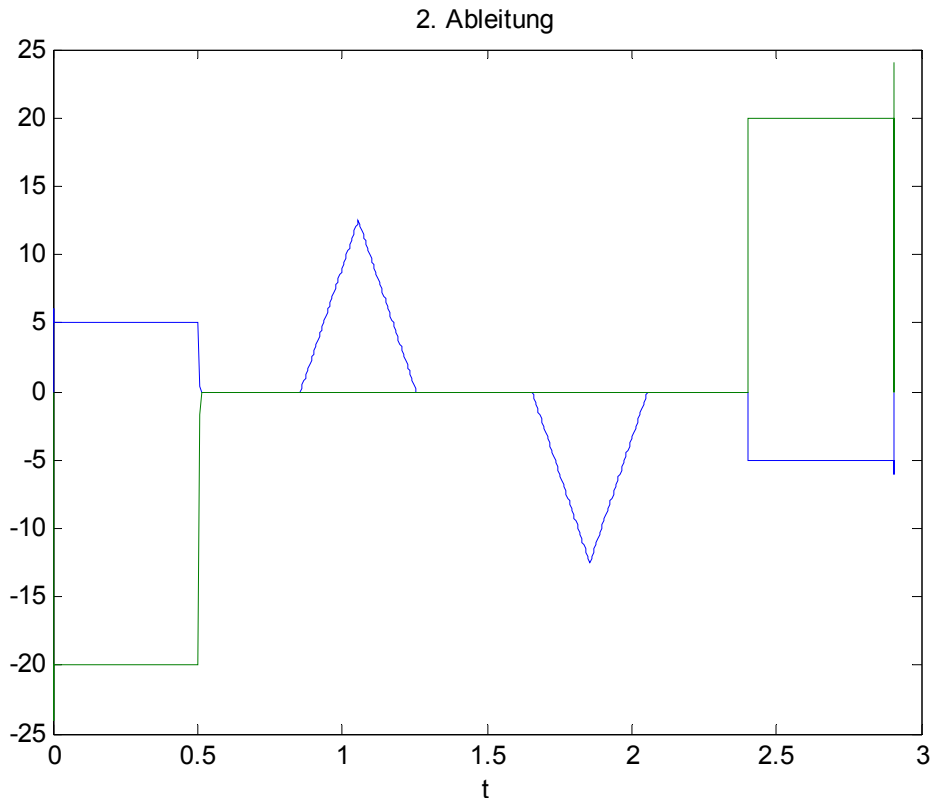


Abbildung 5.20: Gelenkwinkelbeschleunigungen nach der Skalierung

Sehr gut zu erkennen ist, wie sowohl die maximale Beschleunigung als auch die maximale Geschwindigkeit ausgenutzt werden. Beide Grenzen werden von der grünen Achse voll ausgenutzt, die blaue Achse folgt den Bewegungen synchron. Wie bei der Zeit $t = 0$ und $t = 2,9$ zu erkennen ist, ergeben sich bei der Beschleunigung der Achsen Überschwinger. Diese sind auf die noch nicht optimale Begrenzung von \ddot{s} zurück zu führen.

5.3 Filter

Für die Implementierung des Filters sind keine besonderen Vorgehensweisen nötig. Die im Kapitel 3.2 vorgestellten Formeln können alle ohne größere Probleme in Programmiersprache umgesetzt werden. Es ist lediglich darauf zu achten, dass zu jedem Abtastzeitpunkt des Filters auch ein Wert am Eingang anliegt. Prinzipiell ist eine sehr kurze Abtastzeit zu empfehlen, da das Filter dann das beste Ergebnis liefert. Besonders das Ausgangssignal wird dadurch sehr glatt und besitzt wenige, und wenn, dann vernachlässigbar kleine Überschwinger.

Soll das Filter in das Programm mit eingebaut werden, so ergibt sich der folgende Programmaufbau des Trajektorien-Generators.

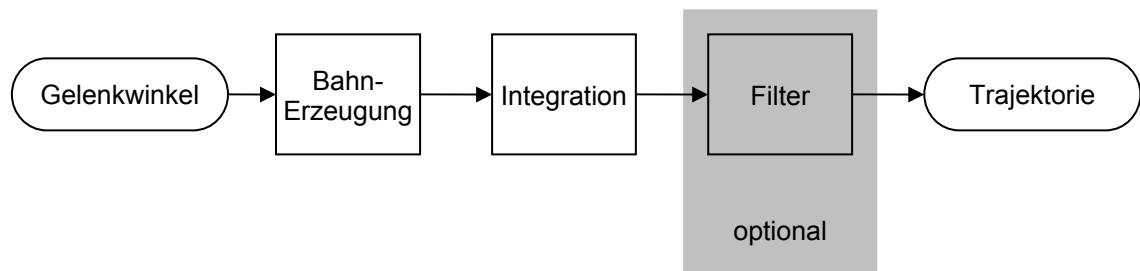


Abbildung 5.21: Aufbau des Trajektorien-Generators

Wird das Filter so wie dargestellt in das Programm mit integriert, so ist es nun möglich den Ruck der geplanten Trajektorie zu begrenzen. Die Begrenzung des Rucks erfolgt allerdings auf Kosten der Genauigkeit. Die so ermittelte Trajektorie weist eventuell größere Abweichungen vom geplanten Pfad auf als zulässig.

6 Zusammenfassung und Ausblick

Mit dieser Bachelorarbeit wurde ausführlich auf die Generierung von synchronisierten Trajektorien aus geplanten Via-Punkten eingegangen und derzeit übliche Lösungsmethoden vorgestellt. Besonders der Ansatz der entkoppelten Trajektorienplanung liefert eine sehr gute Möglichkeit zur Lösung der Aufgabe.

Mit dem vorgestellten Algorithmus konnte der theoretische Ansatz der entkoppelten Trajektorienplanung bereits für Versuchszwecke umgesetzt werden. Das in Kapitel 4.3.2 beschriebene Vorgehen eignet sich also prinzipiell sehr gut für die gegebene Problemstellung. Allerdings sind noch einige Optimierungsaufgaben zu erledigen. Außerdem müssen alle erzeugten Werte noch so angepasst werden, dass sie zu jeder Millisekunde vorliegen. Des Weiteren wäre es wünschenswert, wenn mit diesem Ansatz der Ruck direkt begrenzt werden könnte. Der Ruck kann durch das Filter zwar begrenzt werden, allerdings ergeben sich dadurch unvorhersehbare Abweichungen beim Gelenkwinkel.

Für den nächsten Schritt wäre eine Verbesserung des Algorithmus notwendig. Außerdem ist es als sinnvoll zu erachten das komplette Programm in der Programmiersprache C zu schreiben, da der Code so mit realen Robotern ausgeführt werden kann. Dadurch könnten die Ergebnisse validiert werden.

Literaturverzeichnis

- [1] Bagioti, Luigi; Melchiorri, Claudio: *Trajectory Planning for Automatic Machines and Robots*. Springer-Verlag, Berlin Heidelberg, 2008
- [2] Bezierkurven:
<http://olli.informatik.uni-Idenburg.de/Grafiti3/grafiti/flow10/page7.html>
(Stand: 15.12.2010)
- [3] Bobrow, J. E.; Dubowsky, S.; Gibson, J. S.: *Time-Optimal Control of Robotic Manipulators Along Specified Paths*. In: *The International Journal of Robotics Research*, 1985
- [4] B-Spline-Kurven:
https://www.ads.tuwien.ac.at/docs/lva/mmgdv/k1___014.htm
(Stand: 15.12.2010)
- [5] Choset, Howie; Lynch, Kevin M.; Hutchinson, Seth; u. a.: *Principles of Robot Motion: Theory, Algorithms, and Implementation* / Latombe, Jean-Claude (Vorw.). Verlag: The MIT Press, 2005
- [6] Craig, John J.: *Introduction to Robotics: Mechanics and Control*. 3. Auflage, Verlagsgruppe Pearson Education, 2003
- [7] Kröger, Torsten: *On-Line Trajectory Generation in Robotic Systems: Basic Concepts for Instantaneous Reactions to Unforeseen (Sensor) Events*. Springer-Verlag, Berlin Heidelberg, 2010 (star Reihe, Band 58)
- [8] Pietsch, Ingo T.: *Adaptive Steuerung und Regelung ebener Parallel-roboter*. Braunschweig, Technische Universität, Diss., 2003
- [9] Wikipedia: <http://de.wikipedia.org/wiki/Matlab> (Stand: 15.02.2011)

Abbildungsverzeichnis

Abbildung 2.1:	Zusammenhang Gelenkwinkel zur Bewegung im Raum	4
Abbildung 2.2:	Bahnverlauf eines x-y-Portals mit Via-Punkt.....	6
Abbildung 2.3:	Positionsabweichungen durch Toleranzen im Gelenkwinkel	6
Abbildung 2.4:	Zulässiges Toleranzband zum Einhalten einer vorgegebenen Abweichung	7
Abbildung 2.5:	Verschiedene Möglichkeiten für Punkt zu Punkt Verbindungen	7
Abbildung 2.6:	Bahnverläufe mit unterschiedlichen Synchronitäten	10
Abbildung 3.1:	Punkt zu Punkt Verbindung mit Polynome dritten Grades..	14
Abbildung 3.2:	Punkt zu Punkt Verbindung durch Gerade mit parabolischen Enden	17
Abbildung 3.3:	Verbindung mit Via-Punkt mit Polynome dritten Grades.....	20
Abbildung 3.4:	Verbindung von Via-Punkten durch Geraden mit parabolischen Enden	21
Abbildung 3.5:	Zwei Achsen mit unterschiedlichen Grenzwerten	22
Abbildung 3.6:	Abweichung vom Toleranzband bei zwei Polynome dritten Grades.....	24
Abbildung 3.7:	Abweichung vom Toleranzband bei Geraden mit parabolischen Enden	25
Abbildung 3.8:	Abweichung vom Sollverlauf durch unangepasste Werte...	26
Abbildung 3.9:	Iteratives Anpassen an Grenzwerte mit daraus resultierendem Verlauf	27

Abbildung 3.10:	Abweichung von Spline-Polynome vom Toleranzband.....	30
Abbildung 3.11:	Anordnung eines nichtlinearen Filters	31
Abbildung 3.12:	Aufbau eines nichtlinearen Filters.....	32
Abbildung 3.13:	Verlauf des Ausgangssignals mit unterschiedlichen Abtastzeiten.....	35
Abbildung 3.14:	Ausgang bei einem Sägezahnprofil mit unterschiedlichen Abtastzeiten.....	36
Abbildung 3.15:	Ausgang des Filters mit stetigem Eingangssignal	36
Abbildung 4.1:	Erzeugung von Trajektorien durch Skalierung des Pfades .	38
Abbildung 4.2:	Vorgabe des Verlaufs für die Pfadplanung	40
Abbildung 4.3:	Geplanter Pfad im Toleranzband und mit Start von $s = 0$...	41
Abbildung 4.4:	Erzeugter Pfad mit dazugehörigen Ableitungen	42
Abbildung 4.5:	Erzeugung von Trajektorien durch konstante Skalierung des Pfades.....	45
Abbildung 4.6:	Ableitungen der Trajektorie bei konstanter Skalierung	46
Abbildung 4.7:	Verlauf des abschnittsweise festgelegten λ	47
Abbildung 4.8:	Abschnittsweise Skalierung mit fehlerhaften Ableitungen der Trajektorie	47
Abbildung 4.9:	Vorwärts- und Rückwärtsintegration im Zustandsraum [5] .	50
Abbildung 4.10:	Ermittlung der Umschaltunkte um Schneiden der MGK zu verhindern [5].....	52
Abbildung 4.11:	Geplanter Pfad für mehrere Achsen	53
Abbildung 4.12:	Skalierung mit $s(t)$	54
Abbildung 4.13:	Neu ermittelte Trajektorien	54
Abbildung 5.1:	Gelenkwinkel mit Toleranzband.....	57
Abbildung 5.2:	Verbindung der Gelenkwinkel mit Geraden	58

Abbildung 5.3:	Verbindung von zwei Geraden mit Kurvenstück	60
Abbildung 5.4:	Iteratives Verkürzen des Kurvenstücks bis Toleranzband eingehalten wird	61
Abbildung 5.5:	Kurvenstück aus zwei Polynomen dritten Grades	62
Abbildung 5.6:	Nicht synchronisierte Bewegungsphasen bei zwei Achsen	66
Abbildung 5.7:	Voll-Synchrone Bewegungsphasen von zwei Achsen	68
Abbildung 5.8:	Optimierung der Krümmung im Übergang zwischen zwei Bogenstücken	69
Abbildung 5.9:	Verringerung der maximalen Krümmungen durch Anpassung im Übergang	69
Abbildung 5.10:	Neuer Pfad nach Optimierung der Krümmung.....	71
Abbildung 5.11:	Krümmungen nach dem Optimierungsvorgang	71
Abbildung 5.12:	Beispiel eines geplanten Pfads von drei Achsen	74
Abbildung 5.13:	Pfad und Ableitungen neu skaliert, so dass Start bei $s = 0$.	75
Abbildung 5.14:	Beispiel von Pfad und Ableitungen für Integrationsvorgang	75
Abbildung 5.15:	Maximale-Geschwindigkeitskurve im Zustandsraum.....	76
Abbildung 5.16:	Vorwärts- und Rückwärtsintegration im Zustandsraum	77
Abbildung 5.17:	Skalierung mit $s(t)$	78
Abbildung 5.18:	Resultierende Trajektorien nach der Skalierung.....	79
Abbildung 5.19:	Gelenkwinkelgeschwindigkeiten nach der Skalierung	79
Abbildung 5.20:	Gelenkwinkelbeschleunigungen nach der Skalierung	80
Abbildung 5.21:	Aufbau des Trajektorien-Generators	81